Wiesław Gliński

MODEL SIECI INFORMACYJNYCH

WYDAWNICTWO SBP

NAUKA-DYDAKTYKA-PRAKTYKA

MODEL
SIECI
INFORMACYJNYCH

Mojej Matce poświęcam

Polish Librarians Association
SCIENCE-DIDACTICS-PRACTICE

Wiesław Gliński

MODEL OF INFORMATION NETWORKS

WYDAWNICTWO SBP

Warsaw 1999

Stowarzyszenie Bibliotekarzy Polskich
NAUKA-DYDAKTYKA-PRAKTYKA

Wiesław Gliński

MODEL SIECI INFORMACYJNYCH

WYDAWNICTWO SBP

Warszawa 1999

Książka wydana staraniem Stowarzyszenia Bibliotekarzy Polskich przy finansowym wsparciu Instytutu Informacji Naukowej i Studiów Bibliologicznych Uniwersytetu Warszawskiego w ramach badań statutowych

Recenzenci
prof. dr hab. Mieczysław MURASZKIEWICZ
prof. dr hab. Henryk RYBIŃSKI

Redaktor tomu
Janusz NOWICKI

Korekta
Jadwiga KRĘŻLEWICZ

ISBN 83-87629-21-9

CIP — Biblioteka Narodowa
Gliński, Wiesław
Model sieci informacyjnych / Wiesław Gliński ; Stowarzyszenie Bibliotekarzy Polskich. — Warszawa : Wydaw. SBP, 1999. — (Nauka, Dydaktyka, Praktyka ; t. 34)

Wydawnictwo SBP. Warszawa 1999. Wydanie I.
Ark. wyd. 5.0. Ark. druk. 8.75. Łamanie: R. Lipnicki, K. Brawiński
Druk i oprawa: Zakład „PRIMUM" Kozerki 17a, 05-825 Grodzisk Maz.

SPIS TREŚCI

CONTENTS

Od autora

Przedkładając uwadze Czytelnika niniejszą rozprawę pragnę zaznaczyć, że jest ona wynikiem badań przeprowadzonych przeze mnie w 1996 r. w Instytucie Informacji Naukowej i Studiów Bibliologicznych Uniwersytetu Warszawskiego (dawniej Instytut Bibliotekoznawstwa i Informacji Naukowej Uniwersytetu Warszawskiego). Jest już faktem powszechnie znanym, że w dziedzinie zastosowania sztucznych inteligencji w wyszukiwaniu w sieci Internet dokonano od tego czasu wiele, niemal wprost rewolucyjnych zmian. Po kilku latach zaistniały jednak możliwości opublikowania moich prac, z czego postanowiłem skorzystać. Tak więc tekst, który Czytelnik dostaje do swych rąk ma w tej chwili w dużej mierze charakter archiwalny niż odkrywczy, jednak uważam, że część teoretyczna (rozdz. 3. Model samouczącego mechanizmu dostępu do sieci) może nadal pozostawać inspiracją dla osób zajmujących się badaniem sieci informacyjnych.

Pragnę również poczynić trzy uwagi nakreślające przyczyny, które spowodowały napisanie tej pracy, i które wyznaczają szerszy kontekst dla rozważań szczegółowych prowadzonych w dalszych jej partiach.

Uwaga pierwsza jest w gruncie rzeczy tezą, że skuteczne uprawianie tzw. humanistycznych zawodów jest obecnie bardzo utrudnione bez znajomości dzisiejszych środków technicznych przeznaczonych do pozyskiwania, przetwarzania i prezentowania informacji. Olbrzymia ilość danych i informacji tworzonych i dostępnych we współczesnych, demokratycznych i wysokorozwiniętych społeczeństwach sprawiają, że dostęp do nich staje się coraz bardziej utrudniony. Nadmiar informacji i problemy z jej selekcją są równie kłopotliwe jak brak informacji. Z drugiej strony w wielu środowiskach utrzymuje się niechęć do sięgnięcia po te środki, niechęć spowodowana przeświadczeniem, że posługiwanie się współczesnymi narzędziami w procesach informacyjnych jest trudne, dostępne tylko dla specjalistów. Część z tych obaw jest zapewne nie pozbawiona racji, nie wszystkie jednak poddane próbie okazują się uzasadnione. Istnieje bowiem szybko rosnąca grupa narzędzi informacyjno-informatycznych, które już w pierwszym kontakcie, bez specjalnego szkolenia, okazują się łatwe i skuteczne w użyciu. Moim zamiarem było właśnie przyczynienie się do opracowania takiego „przyjaznego" narzędzia, które ułatwiałoby dostęp i poruszanie się w rozległych sieciach informacyjnych.

Uwaga druga dotyczy współczesnych skomputeryzowanych sieci informacyjnych. Otóż w całej rozciągłości sprawdził się slogan jednej z firm: „sieć jest komputerem". Więcej, prawdziwe jest już dziś twierdzenie odwrotne, tzn. „komputer jest siecią". To sieci bowiem sprawiły, a Internet jest tu koronnym przykładem, że z niewielkiego komputera osobistego wyposażonego w modem i połączonego z siecią można wykonywać skomplikowane zadania wymagające ogromnych mocy obliczeniowych na odległych geograficznie komputerach. Dostęp do katalogu bibliotecznego znajdującego się na innym kontynencie jest tak samo prosty jak do katalogu w „mojej" bibliotece. Rozległe sieci informacyjne, które zniosły pojęcie fizycznego dystansu sprawiły, że koncepcja „globalnej wioski" opisanej przez McLuhana stała się rzeczywistością w odniesieniu do przestrzeni informacyjnej. Konsekwencje tego faktu dla pracy humanistów, techników, pracowników administracji i przedstawicieli innych grup zawodowych są bardzo głębokie i rozległe. Naturalnie nie sposób ich tutaj przedyskutować, ani nawet wymienić. Jedno jest wszak pewne – i po-

twierdza to tezę zawartą w uwadze pierwszej – że brak dostępu do sieci informacyjnych i/lub nieumiejętność poruszania się w nich mocno ograniczają możliwości zawodowe, zwłaszcza tych, dla których bogata, różnorodna informacja jest „intelektualnym paliwem".

Uwaga trzecia wiąże się z podejmowaną niekiedy dyskusją na temat „nowe media versus tradycyjne środki przekazu". W takich dyskusjach przeciwstawia się niekiedy komputery i sieci informacyjne książkom. Zobaczmy co w tej sprawie mówi Umberto Eco, włoski semiotyk, eseista i pisarz:

„Powinniśmy nauczyć się używania i Internetu i CD-ROM-ów, by dzięki temu nauczyć naszych bliźnich czytać także książki. To jest możliwe. Nadchodzące czasy zapowiadają człowiekowi kultury nowe obowiązki i nowe doświadczenia. Niegdyś człowiekiem kultury był ten, kto umiał czytać i pisać książki, ale mógł pisać je także odręcznie, powierzając mechaniczną pracę nad nimi swoim sekretarzom albo kopistom. Dzisiaj od człowieka kultury wymaga się znajomości zarówno książek, jak i nowych form pisania i gromadzenia informacji. Tylko w ten sposób można zagwarantować, że nowe media będą używane w sposób demokratyczny, bez odsuwania kogokolwiek od zasobów informacji, tylko tak można uczyć każdego, jak wybierać i jak oceniać informacje, które otrzymuje i jednocześnie utrzymywać przy życiu ten niezbędny instrument rozwoju ludzkiego i kulturalnego, jakim jest książka."[1]

Ten sam autor odnotował, że już obecnie daje się zauważyć podział społeczeństwa na trzy klasy: osoby nie mające dostępu do komputerów (a tym samym często i książek) i uzależnionych prawie całkowicie od przekazu audiowizualnego czyli telewizji, do drugiej klasy należą ci, którzy potrafią korzystać z komputera na poziomie biernym (np. urzędnicy bankowi, pracownicy biur rezerwacji linii lotniczych itp.) oraz na samym szczycie znajdują się osoby potrafiące świadomie i w sposób twórczy korzystać z komputera i sieci informacyjnych dokonując analiz, potrafiące odróżniać informacje wartościowe od bezwartościowych. Ułatwienie dostępu do komputera (który jest siecią!), a to jest właśnie najszerzej rozumiany cel tej pracy, może sprawić, że ta trzecia grupa będzie się stale powiększała.

Podziękowania

Autor pragnie podziękować swojemu promotorowi, prof. dr. hab. M. Muraszkiewiczowi, za pomoc i motywację, bez których nie powstałaby niniejsza praca oraz prof. dr. hab. M. Drzewieckiemu, dyrektorowi Instytutu Informacji Naukowej i Studiów Bibliologicznych Uniwersytetu Warszawskiego, za umożliwienie dostępu do zaawansowanych sieciowych systemów komputerowych i nieustającą zachętę do napisania oraz cenne uwagi, które wykorzystałem w trakcie pisania tej książki. Chciałbym również wyrazić swoje podziękowania prof. dr. hab. Henrykowi Rybińskiemu za cenne wskazówki, które znalazły swój wyraz w tej pracy, pani mgr Bogumile Rykaczewskiej-Wiorogórskiej – głównemu ekspertowi z Centrum Informatycznego Uniwersytetu Warszawskiego (CIUW) za pomoc w uzyskaniu materiałów i dostępu do laboratoriów komputerowych w CIUW; prof. Paolo Tosoliniemu z Uniwersytetu w Trieście (Włochy) za zgodę na wykorzystanie funkcji z jego systemu MMWWWPC (wersja 2.0) napisanych w języku OpenScript; panu Antonowi Reinhardowi (*General Manager Central Europe Asymetrix GMBH*) za nieodpłatne udostępnienie kopii systemu ToolBook.

Warszawa, 20 lutego, 1999 r. Wiesław Gliński

[1] Cytat pochodzi z wykładu pt. „Nowe środki masowego przekazu a przyszłość książki", jakiego udzielił prof. Umberto Eco w PEN Clubie 23 lutego 1996; tłum. Adam Szymanowski.

1. WSTĘP

1.0. SIECI INFORMACYJNE

1.0.1. Zagadnienie sieci informacyjnych

Coraz częściej socjologowie i badacze kultury współczesnej odwołują się do terminu *społeczeństwo informacyjne*. Cechą szczególnie odróżniającą ten rodzaj społeczeństwa od innych typów dużych, zorganizowanych grup ludzkich jest znaczny i stale rosnący wpływ informacji i technik jej przetwarzania na funkcjonowanie takich społeczeństw. Dziś informacja staje się w procesach rozwojowych czynnikiem stawianym na równi z kapitałem, pracą, energią i zasobami naturalnymi.

Jednak sama informacja, bez sprawnych i tanich środków jej przenoszenia i dostępu do niej nie odgrywałaby tak doniosłej roli, jak ma to miejsce obecnie w zaawansowanych technologicznie społeczeństwach. Tym elementem, który sprawia, że informacja może być dostarczona w dowolne miejsce, do dowolnego użytkownika i o dowolnej porze, są sieci komputerowe. I odwrotnie, to użytkownik dzięki sieciom komputerowym, a dokładniej sieciom informacyjnym, może dotrzeć do danych oddalonych geograficznie od niego o setki lub tysiące kilometrów.

Sieci są więc potężnym narzędziem przenoszenia informacji. Dlatego buduje się ich coraz więcej, stają się one coraz bardziej złożone, podlegają procesom integracyjnym, tzn. łączą się w jeszcze większe sieci i przez to w jakiś sposób wychodzą poza środowiska, w których powstały. Przykładem tego zjawiska jest bezprecedensowy rozwój sieci Internet.

Rosnąca skala sieci, czemu towarzyszy nadzwyczajny wzrost liczby źródeł informacji i ilości samej informacji, sprawiają, że z punktu widzenia ich użytkowników głównym problemem staje się opanowanie umiejętności poruszania się w sieci, czyli umiejętności nawigowania oraz wyszukiwania.

Niniejsza praca jest próbą zarysowania teorii i stworzenia systemu ułatwiającego zarówno nawigowanie, jak i docieranie do poszukiwanych informacji w rozległych, niejednorodnych sieciach informacyjnych.

Definicja sieci

Z czysto formalnego i zarazem ogólnego punktu widzenia sieć jest zbiorem pewnych elementów, pomiędzy którymi występują określone relacje (związki). Pomimo dużego poziomu ogólności definicja ta jest pożyteczna – zostanie ona

wykorzystana w rozdziale 3 jako podstawa do budowy modelu formalnego analizowanych tutaj procesów.

Dla zachowania terminologicznej jasności trzeba odnotować, że w tej pracy terminy *dane* i *informacja* uważa się za synonimy.

Systemem informacyjnym nazywa się cztery wzajemnie sprzężone elementy:

- uporządkowany zbiór informacji,
- zestaw procedur operowania na tym zbiorze (np. katalogowanie, wyszukiwanie, przygotowywanie sprawozdań),
- środki organizacyjne i techniczne zapewniające funkcjonowanie systemu (np. urząd, komputery, kserokopiarki),
- zasoby ludzkie zaangażowane w zarządzanie systemem i realizacje jego funkcji.

W przypadku gdy zbiór informacji przechowywany jest na nośnikach komputerowych zaś procedury operowania na tym zbiorze wykonywane są przez sprzęt komputerowy powiada się, że mamy do czynienia z *systemem informatycznym.* Odnotujmy, że ustrukturalizowany zbiór informacji przechowywany w pamięci komputera(ów) nosi nazwę *bazy danych.* Do tych określeń dodajmy, że przez *system komputerowy* rozumiemy sprzęt komputerowy z podstawowym oprogramowaniem umożliwiającym jego funkcjonowanie. Na tym samym sprzęcie komputerowym można realizować jednocześnie więcej niż jeden system informatyczny.

Z podanej wyżej definicji wynika jednoznacznie, że termin „system informacyjny" jest szerszy niż termin „system informatyczny": każdy bowiem system informatyczny jest systemem informacyjnym, ale nie odwrotnie. Niekiedy używa się określenia *skomputeryzowany system informacyjny.* Jest to system informacyjny, w ramach którego zrealizowano system informatyczny.

1.1. TEZA PRACY

Przedstawiając model formalny i model komputerowy wraz z eksperymentem praktycznym udowodniono, że jest możliwe opracowanie samouczącego się mechanizmu do wspomagania nawigacji w rozproszonych, heterogenicznych sieciach informacyjnych w celu uzyskania zasobów relewantnych do zadanego pytania.

1.2. PRZEGLĄD ZAWARTOŚCI PRACY

W rozdziale 2 (Sieci informacyjne) przedstawiono zasady działania złożonych sieci informacyjnych, omówiono największe sieci komputerowe ze szczególnym uwzględnieniem sieci Internet, starano się przy tym wykazać, że rozwój sieci komputerowych zależny jest od rozwoju technologicznego oraz potrzeb użytkowników.

Rozdział 3 (Model samouczącego się mechanizmu dostępu do sieci) omawia model matematyczny samouczącego się mechanizmu dostępu do sieci. Zostało przedstawionych szereg definicji i twierdzeń dotyczących sieci, zleceń i skierowy-

wania zleceń do sieci. Do najważniejszych z nich należy twierdzenie o realizowalności procesu obsługi zlecenia (twierdzenie 3.5).

Rozdział 4 (Eksperyment) opisuje program komputerowy (o nazwie *NetExp*), który jest ilustracją idei modelu matematycznego omówionego w rozdziale 3. System NetExp został stworzony przez autora przy wykorzystaniu narzędzi systemu ToolBook (wersja 3.0) oraz obiektowego języka programowania OpenScript systemu ToolBook.

W rozdziale 5 opisano architekturę środowiska współdziałania z siecią programu NetExp. Eksperyment wykorzystujący system NetExp został przeprowadzony w laboratorium komputerowym IBIN UW (obecnie Instytut Informacji Naukowej i Studiów Bibliologicznych Uniwersytetu Warszawskiego), które posiada dostęp do Internetu.

W rozdziale 4 i 5 odwołano się do szeregu funkcji, których szczegółowy opis znajduje się w zbiorach biblioteki Instytutu Informacji Naukowej i Studiów Bibliologicznych UW.

Do pracy załączony jest również wykaz ważniejszych skrótów i oznaczeń oraz indeks ważniejszych terminów.

Przy pisaniu tej książki natknięto się na szereg problemów terminologicznych. W wielu przypadkach koniecznym było zdefiniowanie własnych terminów, na przykład na określenie miary dokładności zleceń adresowanych do sieci (współczynnik Θ, patrz rozdz. 3.4). Autor ma świadomość, że niektóre terminy ogólne zostały użyte w węższym sensie niż ma to miejsce na gruncie nauk humanistycznych. Przykładem jest tu termin *„uczenie się”*, który w tej pracy ma zakres znaczeniowy węższy niż na gruncie psychologii czy pedagogiki. Tutaj odnosi się on do „maszyny” a nie do człowieka. Uczenie jest rozumiane jako systematyczne, ustrukturalizowane powiększanie wiedzy o zasobach i sposobie organizacji sieci informacyjnych.

2. ORGANIZACJA SIECI INFORMACYJNYCH

2.0. WSTĘP

Rozdział ten poświęcony zostanie przede wszystkim zagadnieniom organizacji sieci informacyjnych. Sieci informacyjne rozpatrywane w tej pracy oparte są na sieciach komputerowych, dlatego zostanie także przedstawiona zasada działania dużych, złożonych sieci komputerowych; za podstawę rozważań będzie przyjęty model ISO/OSI (ang. *International Standards Organization Open Systems Interconnection Reference Model*). Szczególny nacisk będzie położony na sieć Internet oraz na wykorzystanie inteligentnych systemów wyszukiwawczych. Istotną tezą tego rozdziału jest stwierdzenie, że sieć, zarówno komputerowa, jaki i informacyjna, ewoluuje na skutek – z jednej strony – rozwoju środków technicznych, zwłaszcza telekomunikacji oraz – z drugiej strony – w wyniku żądań i potrzeb użytkowników.

2.1. SIECI

Z ogólnego, abstrakcyjnego punktu widzenia sieć określa się jako „złożony system stworzony ze wzajemnie połączonych i rozproszonych w przestrzeni połączeń, podsystemów, i/lub elementów"[2]. Zwykle w opisie topologii sieci mówi się o węzłach i łączących je połączeniach, które stanowią materializację relacji występujących pomiędzy węzłami. Dokładniejszą definicję sieci podamy w rozdz. 3.

W ogólnosystemowej literaturze[3] na temat sieci odróżnia się sieci na poziomie makro od sieci na poziomie mikro. Do typowych przykładów sieci w skali makro zalicza się m.in. drogi kolejowe, trasy lotów lotniczych, zaś w skali mikro m.in. układ krążenia i sieci neuronowe. Ponadto rozróżnia się charakter dynamiczny i statyczny sieci. Warto odnotować, że wymienione powyżej przykłady sieci mają charakter dynamiczny, podczas gdy tezaurusy, systemy klasyfikacji czy plany mają charakter statyczny.

Zwraca się uwagę na fakt, że rozwój sieci o zasięgu globalnym powinien być motywowany następującymi czynnikami:[4]

---

[2] Wg [BOR77], por. [LAN76]

[3] Np. [BOR77], por. [BEC73]

[4] Wg [BOR77], por. [BEC73]

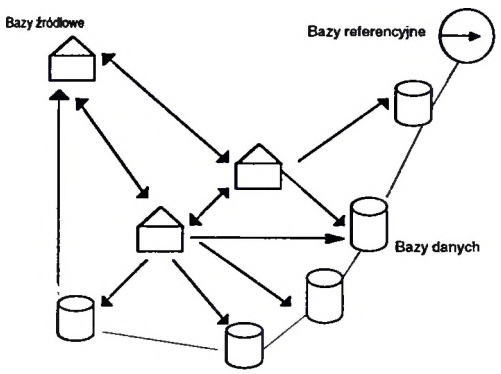
- tworzenie symetrycznych połączeń komunikacyjnych,
- włączanie do sieci elementów odizolowanych lub całych systemów,
- dzielenie zasobów i kosztów ich utrzymania,
- decentralizacja i redystrybucja usług i korzyści wynikłych z sieci informacyjnych,
- zwiększanie niezawodności i bezpieczeństwa zasobów informacyjnych,
- zmniejszanie strat czasu i energii koniecznych do utrzymania systemów,
- wzajemna wymiana doświadczeń,
- zachowanie równowagi w kontroli systemu.

Wśród istotnych czynników mających wpływ na rozwój sieci wymienia się pięć istotnych elementów:

- ludzie – elementem kluczowym w każdym systemie jest: personel i możliwości pracowników,
- maszyny – (w tym komputery i inne środki techniczne),
- zasoby finansowe,
- metody – procedury,
- pomiary w celu testowania systemów i badania różnych parametrów eksploatacyjnych.

2.2. SIECI INFORMACYJNE

Ogólną definicję sieci informacyjnej zawarliśmy w poprzednim podrozdziale. W definicji tej kluczową rolę odgrywają zasoby informacyjne. Teraz podkreślimy, że zasoby te często dzieli się na trzy podstawowe elementy, co ułatwia zarządzanie sieciami (patrz rys. 2.1):



Rys. 2.1 Sieć informacyjna[5]

- bazy źródłowe (ang. *Masterbases*) – zbiory pełnych informacji. Czas dostępu do zbioru jest wolny, ale koszty składowania niskie;

[5] Wg [BOR77].

- bazy danych (ang. *Databases*) – zdecentralizowane zbiory danych utworzone na podstawie zbiorów podstawowych;
- bazy referencyjne (ang. *References*) – wyspecjalizowane bazy, które skierowują do odpowiednich baz źródłowych i baz danych.

Z praktycznego punktu widzenia trzeba odnotować, że rozwinięte sieci informacyjne obejmują różnego rodzaju dane: mikrofilmy, taśmy z nagraniami, obrazy, grafikę, filmy, dane tekstowe, słowniki, tezaurusy itp., zlokalizowane w rożnych instytucjach rozproszonych geograficznie. Obecnie na określenie niejednorodnych zasobów informacyjnych, w skład których wchodzi informacja w postaci graficznej, tekstowej i/lub dźwiękowej często stosuje się termin *bazy multimedialne* (rys. 2.18)[6].

2.3. SIECI KOMPUTEROWE

Kośćcem współczesnych sieci informacyjnych są sieci komputerowe, które stanowią fizyczną platformę przechowywania, przetwarzania i przesyłania informacji. Węzłami sieci komputerowych są komputery, połączenia zaś to linie telekomunikacyjne do przesyłania danych. Początkowo sieci komputerowe miały prostą, najczęściej gwiaździstą strukturę, zaś celem ich funkcjonowania było zapewnienie szybkiego dostępu wielu rozproszonych użytkowników do zasobów obliczeniowych sieci. Powoli struktura sieciowych systemów komputerowych ewoluowała w stronę struktury drzewiastej, czy połączenia struktury drzewiastej ze strukturą gwiazdy. Stawało się opłacalnym łączenie geograficznie odległych sieci komputerowych o topologii gwiazdy w większe całości. Tak właśnie powstały współczesne sieci komputerowe.

W sieciach komputerowych wyróżnia się *podsystem transportowy* odpowiedzialny za przesyłanie informacji oraz *podsystem zarządzania* sterujący zasobami sieci. Podsystem transportowy jest częścią sieci telekomunikacyjnej, czyli zespołu urządzeń i procedur umożliwiających użytkownikowi wykorzystanie dostępnych zasobów. Do typowych funkcji sieci telekomunikacyjnej zalicza się[7]:

- udostępnianie ścieżek dostępu,
- konwersję informacji przedstawionej w postaci cyfrowej na sygnał przesyłany za pomocą medium transmisyjnego (sygnał elektryczny, optyczny, radiowy itd.),
- grupowanie informacji w ramki, pakiety lub komunikaty umożliwiające ich niezawodną transmisję,
- wybór drogi przesyłania informacji oraz diagnostykę linii przesyłu danych.

W zależności od sposobu połączenia sieci klasyfikuje się następująco[8]:

---

[6] Por. podroz. 2.6, gdzie zostanie przedstawiona nieco szerzej charakterystyka hipermedialnych (multimedialnych) systemów informacyjnych.

[7] Wg [PAS95].

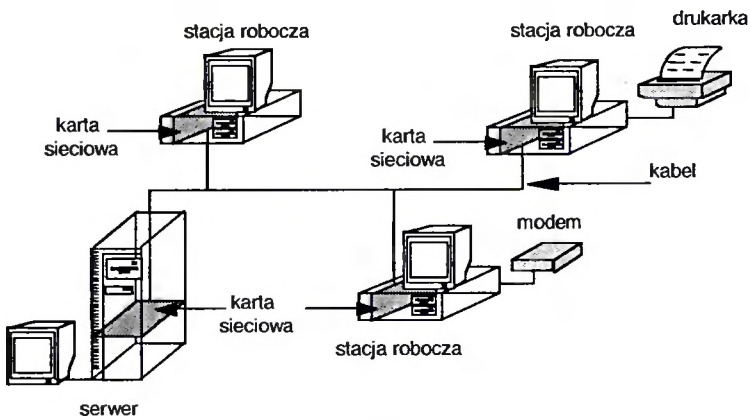
[8] Wg [PAS95].

- wielodostępne – informacja przesyłana między parą komputerów dostępna jest dla wszystkich użytkowników,
- komutowane – przed rozpoczęciem transmisji danych nawiązywane jest połączenie między nadawcą i odbiorcą informacji,
- hybrydowe – są połączeniem sieci komutowanych i wielodostępnych.

Zwraca się uwagę na kilka powodów tworzenia sieci komputerowych:[9]

- współużytkowanie programów, baz danych i pozostałych zasobów sieci; do zasobów zalicza się przeważnie drukarki, plotery oraz urządzenia pamięci masowej; bazy danych dostępne przez sieć komputerową stwarzają możliwość blokowania rekordu, co pozwala na jednoczesny dostęp do rekordu (bez jego niszczenia) wielu użytkownikom,
- ograniczenie wydatków na zakup komputerów,
- grupy robocze; dzięki istnieniu odpowiedniego oprogramowania do funkcjonowania grup dyskusyjnych możliwy jest swobodny obieg informacji i dokumentów między członkami grup dyskusyjnych,
- poczta elektroniczna,
- ułatwienie zarządzania zbiorami,
- rozwój organizacji.

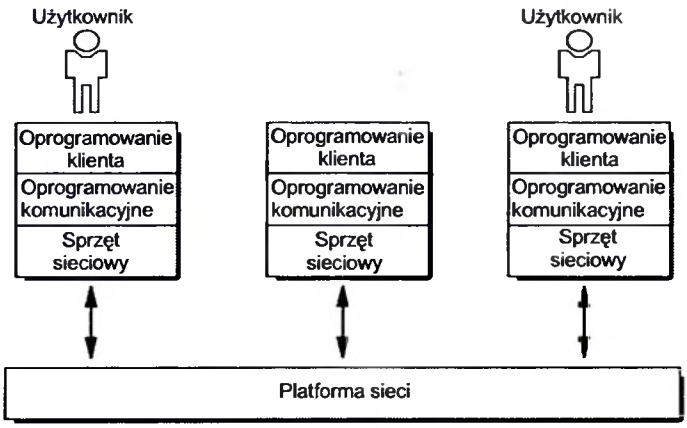
Podamy teraz dokładniejszą definicję sieci komputerowej. *Sieć komputerową* określa się jako system komunikacyjny służący przesyłaniu danych i łączący komputery i urządzenia peryferyjne[10]. Jest rzeczą interesującą, jak ważne jest medium transmisyjne[11] (jako podstawowy nośnik danych), łączący składniki sieci (łączy on karty sieciowe zainstalowane w każdym z urządzeń należących do sieci) – zalicza się go do podstawowych komponentów sieci (patrz rys. 2.2).



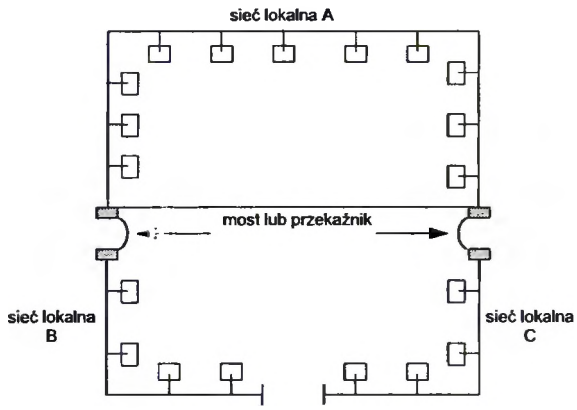
Rys. 2.2 Sieć komputerowa i jej składniki

[9] Wg [TAN89], [SHE95].
[10] Wg [SHE95].
[11] Np.: kabel, fale radiowe, podczerwień itd.

W strukturze logicznej sieci komputerowej wyróżnia się oprogramowanie użytkownika (niekiedy nazywanego klientem), wykorzystujące oprogramowanie komunikacyjne sieci do wymiany informacji z innymi użytkownikami sieci lub z komputerami bazowymi (patrz rys. 2.3).



Rys. 2.3 Komunikacja w sieci[12]



Rys. 2.4 Sieci lokalne i sieć złożona

Ze względu na rozmiary sieci (liczbę komputerów, zasięg terytorialny) można rozróżnić sieci lokalne (ang. *Local Area Network*) i sieci rozległe (ang. *Wide Area Network*). Niezależnie od tego podziału istnieją sieci złożone łączące w jedną całość kilka sieci lokalnych lub kilka sieci rozległych[13].

Sieci lokalne połączone w jedną sieć złożoną[14] umożliwiają większej liczbie użytkowników wzajemną komunikację i współużytkowanie zasobów. Jednak ze

[12] Wg [SHE95].
[13] Wg [SHE95], [GIB93].
[14] Por. rys. 2.4 wg [SHE95].

względu na niewielkie rozmiary taka sieć jest nadal traktowana jako lokalna. Rysunek 2.4 przedstawia przykład lokalnej sieci złożonej.

Termin *sieć złożona* nie jest ściśle zdefiniowany. Intuicyjnie za sieć złożoną uważa się sieć utworzoną z co najmniej dwóch, połączonych sieci. A zatem połączone dwie sieci lokalne wykorzystujące różne protokoły komunikacyjne oraz różne sieciowe systemy operacyjne mogą stanowić jedną złożoną sieć lokalną, lub sieć rozległą, gdy znajdują się w różnych strefach geograficznych (rys. 2.5). Dalej przyjmiemy, że *sieć rozległa* to taka sieć złożona, której używa się do połączenia poszczególnych swoich części publicznych łącz telefonicznych lub innego rodzaju usług telekomunikacyjnych[15] (por. rys. 2.5).

Rys. 2.5 Schemat rozległej sieci komputerowej[16]

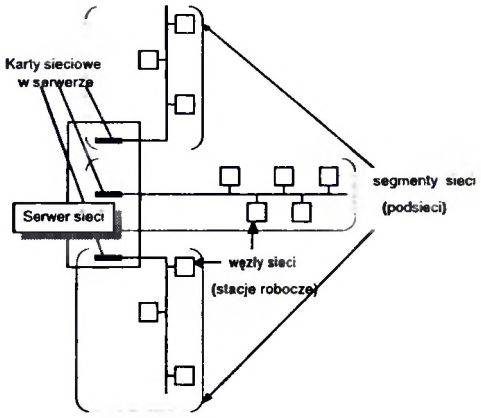
Nieco inna klasyfikacja sieci, oparta na kryterium zasięgu terytorialnego wymienia następujące typy sieci..

- *Segment sieci,* który jest zwykle definiowany przez sprzęt lub specyficzny węzeł sieci[17], np. w środowisku Novell Netware każda karta sieciowa zainstalowana w serwerze tworzy odrębny segment sieci. (rys. 2.6.);
- *Sieć lokalna* (ang. *Local area network – LAN*), określa się jako segment sieci z dołączonymi stacjami roboczymi i serwerami lub też jako połączone sieci lokalne w obrębie tej samej strefy (np. budynku);
- Sieć terytorialna (ang. *Campus network*) obejmuje swoim zasięgiem kilka budynków znajdujących się na terenie uczelni lub przedsiębiorstwa;
- *Miejska sieć komputerowa* (ang. *Metropolitan area network – MAN*) swoim zasięgiem obejmuje miasto wykorzystując do połączenia istniejące rozwiązania telekomunikacyjne;

[15] Wg [SHE95].

[16] Wg [SHE95].

[17] Np. w środowisku sieci Ethernet segmentem sieci jest liniowy odcinek kabla pełniący rolę magistrali. Sygnały wysyłane w tym segmencie odbierane są przez włączone do niego stacje. Jeśli segment połączony jest z innymi segmentami za pomocą mostu lub routera (co umożliwia przesyłanie informacji między innymi segmentami), to mamy do czynienia z siecią złożoną.



Rys. 2.6 Schemat sieci lokalnej w Novell Netware z kilkoma segmentami sieci (podsieciami) wg [SHE95]

- *Sieć rozległa* (ang. *Wide area network – WAN*) i sieć globalna obejmują swoim zasięgiem miasta, kraje i kontynenty. Sieci tego typu wykorzystują istniejące połączenia telekomunikacyjne (podobnie jak sieć typu MAN);
- *Sieć korporacyjna* (ang. *Enterprise network*) łączy systemy wewnątrz organizacji bez względu na istniejące protokoły komunikacyjne, różnice oprogramowania, sieciowe systemy operacyjne czy położenie geograficzne. Siecią korporacyjną może być więc sieć typu LAN, MAN czy WAN. Sieć korporacyjna jest przeważnie systemem rozproszonym, w którym zasoby są ulokowane w całej sieci.

Środowiska sieci

Przez środowisko sieci rozumie się sieciowy system operacyjny oraz protokoły komunikacyjne zapewniające komunikację oraz usługi sieciowe. Wyróżnia się dwa rodzaje sieciowych systemów operacyjnych:

- systemy partnerskie (każdy z każdym, ang. *peer to peer*), pozwalające na wzajemne udostępnianie zasobów komputerów użytkowników. Oznacza to, że żaden system nie jest podporządkowany innemu[18],
- systemy dedykowane (dedykowany serwer), umożliwiają istnienie komputerów pełniących rolę wyłącznie dedykowanego serwera [19].

W sieciach komputerowych tzw. klienci posiadają dostęp do programów i/lub plików w centralnym serwerze (dedykowany serwer) lub w serwerach równorzędnych (każdy z każdym), wykonanie jednak samego programu odbywa się w stacji roboczej, która musi dysponować odpowiednią mocą obliczeniową[20].

Sieci komputerowe można określić też jako *system przetwarzania wsadowego,* w którym wiele serwerów i stacji roboczych zajmuje się przetwarzaniem da-

[18] Np. SNA firmy IBM, UNIX z protokołami TCP/IP, Windows NT.

[19] Np.: Novell Netware z protokołem SPX/IPX.

[20] Por. omówioną w podrozdz. 2.5 architekturę klient-serwer.

nych. Sieci, które rozdzielają przetwarzanie między stację roboczą i serwer są środowiskiem typu klient-serwer[21].

Składniki sieci

Jak wspomniano na początku tego podrozdziału sieć komputerowa składa się z odpowiedniego oprogramowania (system operacyjny serwera, protokoły komunikacyjne, sterowniki kart sieciowych) oraz sprzętu (np.: karty sieciowe wraz z łączącym je kablem).

W sieciach typu partnerskiego (ang. *peer to peer*) każdy węzeł sieci (tj. drukarka, stacja robocza, serwer itp.) współpracuje z systemem operacyjnym, posiadającym usługi sieciowe pozwalające użytkownikowi na współużytkowanie plików i urządzeń oraz funkcje związane z bezpieczeństwem i zarządzaniem danymi. W przypadku zastosowania dedykowanego serwera, sieciowy system operacyjny zainstalowany jest w serwerach, natomiast w stacjach roboczych zainstalowana jest tylko jego część, tzw. część klienta. Ponadto, w zależności od systemów sieciowych, możliwe jest zainstalowanie odrębnych, wyspecjalizowanych serwerów np.: serwera plików, poczty elektronicznej, komunikacyjnego, baz danych, archiwizującego, CD-ROM itd.

Systemy klienta zainstalowane są przeważnie w stacjach roboczych, które podłączone są do sieci za pośrednictwem kart sieciowych. Również system operacyjny, pod kontrolą którego pracuje stacja robocza, może zawierać konieczne oprogramowanie do obsługi kart sieciowych. Oprogramowanie to skierowuje żądania pracujących w sieci użytkowników lub programów do serwera bądź serwerów. W podrozdz. 2.5. (*Sieć Internet*) – architektura klient-serwer zostanie omówiona nieco szczegółowiej.

Karty sieciowe zaprojektowane są dla specyficznego typu sieci np.: Ethernet, Token Ring, FDDI, ARCNET itd. Karty działają w warstwie łącza fizycznego[22] i zapewniają przyłączenie specyficznego typu kabla (np.: kabel koncentryczny, skrętka, światłowód itd.). Karty sieciowe definiują interfejs mechaniczny (warunki podłączenia do kabla) i elektroniczny (metody transmisji strumienia bitów przez media transmisyjne) i kontroli sygnałów zapewniających synchronizację transferu danych przez sieć[23].

Systemy okablowania sieci stanowią medium transmisyjne łączące stacje robocze i serwery. W przypadku sieci bezprzewodowych (na podczerwień, fale radiowe itp.) kable nie są oczywiście wymagane.

Metody łączenia systemów w sieć

Do stworzenia sieci komputerowej konieczne jest połączenie wspomnianych wyżej elementów w sieć, dlatego odróżnia się różnego typu sieci.

[21] Por. omówioną w podrozdz. 2.5 architekturę klient-serwer.
[22] Por. model ISO/OSI.
[23] Wg [SHE95].

Typy sieci

Sieci definiowane są m.in. przez rodzaj użytego kabla[24], jego układ (topologię), wielkość transferu danych, protokoły komunikacyjne, oraz metody dostępu do sieci. W przypadku sieci złożonej stosuje się mosty (ang. *bridge*), przekaźniki (sprzęgi międzysieciowe, ang. *routery*) i bramy (ang. *gateway*). Najpopularniejszymi obecnie typami sieci są: Ehternet (wykorzystujący kabel koncentryczny lub skrętkę), Token Ring, ARCNET, FDDI[25]. Jednym z najważniejszych parametrów jest wielkość i szybkość transferu danych[26]. Mosty i przekaźniki umożliwiają łączenie różnych lub podobnych podsieci (segmentów sieci).

Architektura sieci

Istotną cechą każdej sieci jest jej architektura, która definiowana jest przez topologię, metodę dostępu do okablowania oraz stosowane protokoły komunikacyjne. Uzyskanie dostępu do systemu okablowania danej sieci winno być poprzedzone ustanowieniem sesji z innymi węzłami sieci (stacjami roboczymi, serwerami itd.). Metoda dostępu do systemu okablowania definiuje sposób dostępu do współużytkowanych mediów transmisyjnych, co umożliwia transmisję danych. Protokoły są odpowiednimi regułami i procedurami wykorzystywanymi do komunikowania się przez sieć z innym systemem (systemami).

Topologia

Topologia sieci jest kolejną istotną cechą identyfikującą daną sieć. Jest ona określana, jako mapa przebiegu kabla sieciowego. Wyszczególnia się topologię szyny (magistrali, liniową), pierścienia i gwiazdy, drzewa, łańcucha (rys. 2.7).

Topologia magistrali (liniowa, szyny) obecnie jedna z najpopularniejszych, używa wspólnego kabla, do którego mają dostęp wszystkie przyłączone komputery. Każda jednostka w topologii magistrali przesyła dane bezpośrednio do danego komputera. Istotną zaletą tej topologii jest stosunkowo niewielka ilość kabla, wadą konieczność konkurowania każdej stacji o dostęp do kabla.

W topologii pierścienia dane są przesyłane od jednego komputera do drugiego w pierścieniu, dopóki nie osiągną swojego przeznaczenia.

W topologii drzewa w każdym punkcie podziału (rozgałęzienia) komputer rozsyła sygnały. Ze względu na dość złożoną strukturę utrudnione jest znajdywanie błędów[27].

W topologii gwiazdy wszystkie wiadomości przesyłane są przez centralny komputer. Zaletą tego rozwiązania jest fakt, że każde połączenie nie obsługuje wielu komputerów współzawodniczących o dostęp do kabla, wadą konieczność

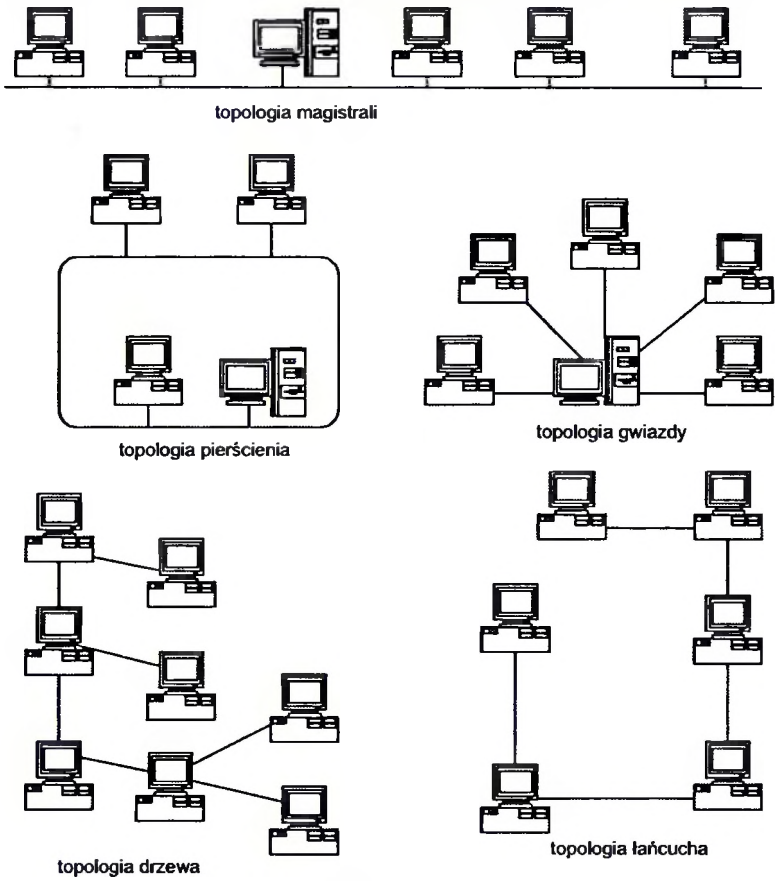
[24] Nie dotyczy to oczywiście sieci bezprzewodowych.

[25] Ang. *Fiber Distributed Data Interface.*

[26] Np. ARCNET pozwala na osiągnięcie 2 Mbit/sec, Token Ring 4 i 16 Mbit/sec, Ehternet 10 i 100 Mbit/sec a FDDI 100 Mbit/sec.

[27] Wg [GIB93] [SHE95].

zainstalowania szybkiego, centralnego komputera, o dużych mocach obliczeniowych, pośredniczącego w przesyłaniu danych między pozostałymi stacjami.



Rys. 2.7 Topologie sieci

Topologia łańcucha jest analogiczna do topologii pierścienia z przerwanym połączeniem. Zaletą tej topologii jest łatwość okablowania, jednak awaria jednego komputera powoduje przerwanie sieci na dwie części[28].

Topologie mogą być jednak mieszane, coraz powszechniejszą topologią staje się tzw. łańcuch gwiazd[29].

Metody dostępu do medium transmisyjnego

Metoda dostępu do okablowania opisuje sposób uzyskiwania dostępu do mediów transmisyjnych przez stację roboczą. W momencie uzyskania dostępu do systemu okablowania, karta sieciowa rozpoczyna wysyłanie pakietów informacji sformatowanych jako strumień bitów.

[28] Wg [GIB93].
[29] Wg [GIB93].

Liniowe systemy okablowania, jak Ehternet, wykorzystują metody dostępu do okablowania. Określane są jako *nasłuchiwanie nośnej, wykrywanie kolizji.* Oznacza to, że stacje robocze po uzyskaniu dostępu do kabla rezygnują z niego, jeśli koliduje to z dostępem innej stacji roboczej. Stacja robocza transmituje sygnał w trybie rozgłaszania tak, że jest on odbierany przez każdą stację, jednak wywołuje reakcję tylko w tej stacji, do której był adresowany. Jeśli obie stacje jednocześnie wysyłają komunikat, dochodzi do kolizji i rozgłaszanie zostaje przerwane.

W przypadku sieci o strukturze pierścienia stosowany jest przeważnie tzw. znacznik, który pozwala stacji na wysłanie komunikatu.

Protokoły komunikacyjne

Protokoły komunikacyjne zarządzają dwoma różnymi poziomami komunikacji. Protokoły *„wysokiego poziomu"* określają sposób komunikacji aplikacji, zaś protokoły *„niskiego poziomu"* określają sposób transmisji sygnałów przez medium transmisyjne. Między tymi poziomami istnieją protokoły zarządzające sesjami komunikacyjnymi między komputerami oraz kontrolujące przepływ informacji w celu wykrycia ewentualnych błędów.

Urządzenia sieciowe

Oprócz kart sieciowych oraz kabla często wymagane są w sieci dodatkowe urządzenia pozwalające m.in. na zwiększenie zasięgu sieci czyli tzw. wzmacniacze (ang. *repeater*), które regenerują sygnały elektryczne i podwajają maksymalną słyszalność sygnału.

Oddzielne segmenty sieci łączone są przy użyciu tzw. mostów (ang. *bridge*). Most może być osobnym urządzeniem lub stanowić część serwera(ów). Np. w serwerze NetWare Novell most jest tworzony przez instalację dwóch kart sieciowych. Mosty przesyłają przez łącze do innych segmentów sieci tylko te informacje, które nie są przeznaczone dla lokalnych węzłów.

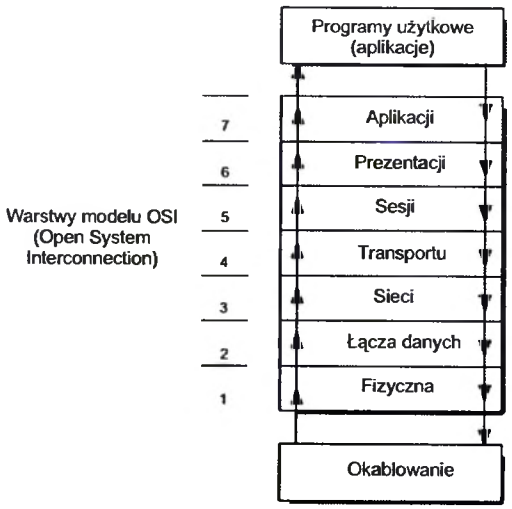
Rozgałęźnik (sprzęg międzysieciowy, ang. *router*) kontroluje przepływ informacji między sieciami. „Odczytuje" adres otrzymanej wiadomości i ignoruje ją, jeśli nie jest to znany adres, który może być osiągnięty przez router. Istnieją dwa rodzaje rozgałęźników: lokalne i zdalne, które z kolei dzielą się na dwa rodzaje:

- rozgałęźniki statyczne, zarządzające siecią, które mają zapisane informacje jakie adresy znajdują się w jakich sieciach,
- rozgałęźniki dynamiczne, przeglądają wszystkie swoje interfejsy i budują tablice, które lokalizują adresy z poszczególnych sieci. Ponadto dynamiczne rozgałęźniki są w stanie wybrać najbardziej optymalną drogę dla danego adresu.

Rozgałęźnik może być ustawiony na selektywne przekazywanie informacji, co pozwala na efektywne odizolowanie wadliwie działających podsieci. Rozgałęźnik oferuje także korzyści związane z bezpieczeństwem danych, blokując przesyłanie niepożądanych danych między komputerami[30].

[30] Por. [SHE95], [GIB93] oraz [PIW95].

Model odniesienia ISO/OSI



Rys. 2.8 Model OSI (Open System Interconnection)

Do właściwego zrozumienia pojęć związanych z systemami sieciowymi konieczne jest przynajmniej pobieżne zapoznanie się z modelem ISO/OSI (ang. *International Standards Organization Open Systems Interconnection Reference Model*). Należy pamiętać, że model ten określa, co ma być wykonane, a nie w jaki sposób. Innymi słowy szczegóły realizacyjne pozostawione są producentom poszczególnych systemów sieciowych. Nie oznacza to jednak, że wszyscy stosują się do modelu OSI, jednak jest on coraz częściej stosowanym modelem odniesienia.

Rysunek 2.8 przedstawia budowę modelu ISO/OSI, który składa się z siedmiu warstw, każda z nich definiuje zbiór usług i związanych z nimi protokołów. Każda z warstw otrzymuje informację bezpośrednio z niższej warstwy dokładając do niej własne usługi. Ponieważ zdefiniowano wiele różnorodnych czynności dla każdej warstwy, model ten można dopasować do szerokiej gamy sprzętu i oprogramowania sieciowego. Model ten ma również znaczenie komercyjne, gdyż wiele rządów tworzy swoje komunikacyjne standardy przy wykorzystaniu modelu OSI.

Warstwa fizyczna

Celem warstwy fizycznej jest dostarczenie danych między urządzeniami sieci. Definiuje ona mechaniczne i elektryczne połączenia z systemem okablowania sieci oraz funkcje zajmujące się właściwą transmisją bitów danych.

Warstwa łącza (łączenia) danych

Warstwa łącza danych określa metody dostępu do okablowania oraz jest odpowiedzialna za wykrywanie kolizji. Warstwa ta definiuje podstawowe jednostki informacji tzw. pakiety oraz metody tworzenia, wysyłania i odbierania pakietów. Zapewnia również bezbłędną komunikację między urządzeniami sieciowymi.

Warstwa sieciowa

Odpowiada za przesyłanie danych między adresami, kontrolowanie przepływu danych oraz łączenie przychodzących pakietów w większe bloki danych i dzielenie bloków na pakiety w celu ich wysłania. Dane mogą być kierowane z jednego rodzaju sprzętu sieciowego do innego[31].

Warstwa transportu

Warstwa ta kontroluje różne procesy sieciowe kierując obsługą sytuacji błędnych (np. zagubionych pakietów)[32]. Warstwa transportowa ma trzy główne funkcje obsługujące transmisję danych do i od warstwy sesji:

- otwieranie połączenia właściwego typu, jakości i prędkości;
- rozpoczynanie transmisji danych i ich zarządzanie;
- zamykanie połączenia.

Warstwa sesji (sesyjna)

Koordynuje współdziałanie między funkcjami i programami użytkowymi wykonywanymi na różnych urządzeniach sieciowych. Warstwa ta określa, który komputer komunikuje się z którym, kontroluje przepływ danych i jest odpowiedzialna za naprawę błędów. Może również pełnić dodatkowe funkcje zarządzające w sieci[33].

Warstwa prezentacji

Określa konwersje kodu i reformatowanie danych dla programów użytkowych[34]. Warstwa zastosowań zajmuje się usługami sieciowymi (np.: związanymi z plikami, wydrukiem, pocztą elektroniczną itp.)

2.4. SIECI KOMPUTEROWE – PRZYKŁADY[35]

Obecnie istnieje wiele sieci komputerowych. Duża część z nich jest sieciami publicznymi, (zarządzanymi przez pocztę, kompanie telefoniczne i telegraficzne), inne mają charakter badawczy (przeważnie są zarządzane i utrzymywane przez wolontariuszy) lub są sieciami komercyjnymi. Poniżej zostaną krótko przedstawio-

---

31 W warstwie tej działają takie protokoły transportu danych jak Internetwork Packet Exchange (IPX) firmy Novell oraz Internet Protocol (IP).

32 W warstwie transportu działa Sequenced Packet Exchange (SPX) firmy Novell.

33 Jak np.: obliczanie należności za korzystanie z sieci.

34 W warstwie tej protokół NetWare File Service Protocol pozwala na konwersje plików pomiędzy formatem plików NetWare Novell (sposób w jaki dane są zapisane na serwerze NetWare Novell) a formatem wymaganym przez inny system operacyjny działający na komputerze użytkownika (DOS, OS/2 itp.).

35 Wg [TAN89].

ne niektóre z sieci komputerowych[36]. Część z nich ma już dzisiaj znaczenie historyczne. Sieci różnią się swoją historią, sposobem zarządzania, oferowanymi usługami, architekturą i grupą końcowych użytkowników. Historia i zarządzanie siecią może różnić się w zależności od tego, czy sieć jest projektowana przez pojedynczą organizację z określonymi celami, czy też dana sieć jest projektowana *ad-hoc*, jako wynik połączenia różnych komputerów bez z góry określonych planów i centralnego zarządzania. Usługi oferowane przez sieci są różne: począwszy od zwykłej komunikacji między dwoma komputerami przez pocztę elektroniczną, transfer plików, zdalną rejestrację (ang. Login) aż po zdalne wykonywanie zadań wsadowych. Techniczna strona projektu różni się w zależności od: rodzaju nośników danych, użytych algorytmów przekazu danych, ilości warstw komunikacyjnych i użytych protokołów. W końcu sama społeczność użytkowników jest różna: począwszy od pojedynczej organizacji a skończywszy na złożonej społeczności naukowców i studentów.

Sieci publiczne

W wielu krajach organizacje rządowe lub prywatne korporacje od jakiegoś czasu zaczęły oferować usługi sieciowe różnym instytucjom. W przypadku sieci publicznych dana podsieć jest własnością operatora dostarczającego usługi komunikacyjne dla grupy komputerów bazowych i terminali. System taki jest określany jako sieć publiczna. Jest on analogiczny do systemu telefonii, a często jest częścią systemu telekomunikacyjnego w danym kraju. Chociaż sieci publiczne w różnych krajach różnią się od siebie wewnętrznie, jednak wszystkie one odwołują się w jakiś sposób do modelu OSI. Warto przy tym zaznaczyć, że wiele prywatnych sieci także używa protokołów OSI, lub planuje go zastosować w przyszłości.

CCITT (*International Telegraph and Telephone Consultative Committee*) wydał wskazania dotyczące obsługi najniższych trzech warstw protokołu OSI (warstwy: fizyczna, sieci, łącza danych) w sieciach publicznych na świecie. Warstwy te określane są jako *X.25*. Warstwa fizyczna protokołu zwana *X.21*, określa interfejs fizyczny, elektryczny i proceduralny między komputerem bazowym a siecią. Obecnie nie wszystkie sieci spełniają te normy w całości, głównie z powodu przekazu analogowego, a nie cyfrowego przez łącza telefoniczne. Warstwa danych posiada dużą liczbę różnorodnych odmian. Wszystkie one są odpowiedzialne za korekcje błędów w przekazie danych między komputerem użytkownika (terminalem) a siecią publiczną.

Większość terminali „nie rozumie" protokołu X.25, dlatego opracowano szereg standardów komunikacyjnych określających sposób komunikacji z sieciami X.25. W konsekwencji są instalowane tzw. czarne skrzynki, do których dołączane są terminale. Te czarne skrzynki w skrócie określane są jako PAD (ang. *Packet*

[36] Większość informacji przedstawionych w tym podrozdziale odwołuje się do pozycji z 1989 r. [TAN89]. Należy jednak pamiętać, że celem tego podrozdziału jest pobieżne zaznajomienie Czytelnika z przykładami sieci komputerowych w aspekcie historycznym.

Assembler Disassembler), a ich funkcje opisane są w dyrektywach CCITT (ang. *International Telegraph and Telephone Consultative Committee*) określanych jako X.3. Został również określony protokół transmisji między terminalami a PAD, nazywany *X.28*. Natomiast odmienny protokół między PAD a siecią określany jest jako *X.29*. Te trzy zalecenia dotyczące komunikacji między terminalem, PAD a siecią określane są jako potrójne X (ang. *triple X*).

Powyżej warstwy sieciowej sytuacja jest bardziej złożona. ISO (ang. *International Standards Organization*) rozwinęła grupę standardów dotyczących definicji usług związanych z warstwą transportu[37] oraz samego protokołu warstwy transportu[38] oraz usług i protokołu związanego z warstwą prezentacji[39].

Wydaje się, że standardy te zostaną przyjęte przez większość sieci publicznych na całym świecie, choć nie jest to tak konieczne, gdyż wiele aplikacji w sieciach nie wykorzystuje w ogóle warstw: sesji i prezentacji. Warstwa aplikacji nie zawiera jednego, lecz cały zbiór protokołów dla różnych aplikacji. I tak np. protokół FTAM (ang. *File Transfer, Access, Management*) określa sposób transmisji, dostępu i manipulacji na odległych zbiorach.

Protokół *MOTIS* (ang. *Message-Oriented Text Interchange Systems*) używany jest dla poczty elektronicznej, podobny jest w swojej budowie do określonych przez CCITT standardów *X.400*. Protokół *VTP* (ang. *Virtual Terminal Protocol*) określa sposoby komunikacji programów z terminalami (np. pełnotekstowe edytory). Protokół *JTM* (ang. *Job Transfer and Manipulation*) jest wykorzystywany do dostarczania zadań do przetwarzania wsadowego (ang. *batch processing*) na odległych dużych systemach komputerowych (ang. *mainframe computers*). Protokół ten może być wykorzystany do przesyłania programów oraz zbiorów z danymi. Oczywiście ciągle definiowane są nowe standardy dla specyficznych aplikacji.

ARPANET (INTERNET)

Sieć *ARPANET* powstała przy współpracy amerykańskiej agencji ds. Obrony (ang. *Advance Research Project Agency of the US Department of Defense*) oraz uniwersytetów amerykańskich i prywatnych korporacji. Badania prowadzone przez te instytucje doprowadziły do stworzenia w grudniu 1969 r. sieci komputerowej składającej się z czterech węzłów. Obecnie sieć ARPANET znana jako Internet (patrz podrozdz. 2.5) dysponuje tysiącami komputerów bazowych rozmieszczonych na całym świecie.

[37] Norma ISO 8072 (za [TAN89]

[38] Normy ISO 8326 oraz ISO 8327 (za [TAN89]).

[39] Normy ISO 8822 oraz ISO 8823 (za [TAN89]).

MAP i TOP

W 1973 r. Robert Metcafle[40] napisał pracę doktorską, w której opisał swoje badania dotyczące sieci lokalnych (*LAN*). Wkrótce potem został zatrudniony w firmie Xerox Corporation, gdzie razem z Davidem Boggsem i z innymi współpracownikami zastosował lokalną sieć *Ethernet* wykorzystując pomysły ze swojej pracy doktorskiej. Ethernet został zastosowany przez wiele instytucji zaś firma Intel wprowadziła do produkcji procesory (kontrolery) wykorzystujące technologie Ethernetu. Wkrótce grupa osób wchodzących w skład komitetu pod auspicjami *IEEE* (ang. *Institute of Electronic and Electrical Engineers*) rozpoczęła prace nad wprowadzeniem standardu dla sieci lokalnych (LAN). Część tego komitetu należąca do GM (*General Motors*) była szczególnie zainteresowana wprowadzeniem sieci LAN. Firma GM w celu wygrania współzawodnictwa z japońskimi konsorcjami samochodowymi pragnęła wprowadzić sieć komputerową łączącą wszystkie swoje filie (biura, fabryki, sprzedawców, podwykonawców itp.). W momencie, gdy klient zamawiał samochód u któregoś ze sprzedawców GM, komputer sprzedawcy wysyłał wiadomość do komputera firmy GM, który z kolei wysyłał odpowiednie zamówienie do swoich podwykonawców na właściwe materiały (np.: stal, gumę, szkło itp.). Istotną częścią sieci komputerowej GM była automatyzacja linii produkcyjnej, w której wszystkie roboty były podłączone do sieci komputerowej. Samochody na linii produkcyjnej przesuwały się w ustalonym tempie i roboty produkcyjne musiały we właściwym czasie „wiedzieć", kiedy rozpocząć montaż kolejnego produktu, dlatego GM uznał koniecznym posiadanie takiej sieci lokalnej, w której znane są z góry granice najdłuższego czasu transmisji. Niestety sieć Ethernet nie dawała takiej możliwości. Zasadniczo Ethernet wymagał od wszystkich komputerów (podłączonych do sieci) nasłuchiwania. W sieci Ethernet, jeśli kabel nie jest obciążony, każda jednostka może przesyłać dane. Jeśli jednak dwie maszyny dokonują transmisji w tym samym czasie, może wydarzyć się kolizja, co powoduje, że obie maszyny wstrzymują transmisję, następnie odczekają pewien losowy odcinek czasu i ponawiają transmisję. Według teorii nie istnieje górna granica czasu, którą dana maszyna musi odczekać zanim wyśle wiadomość. Przyjmując powyższe przesłanki GM stworzył lokalną sieć (LAN) określaną jako *Token bus*.

W tym samym czasie firma IBM zapowiedziała wprowadzenie swojej sieci lokalnej *Token Ring*. Sieć Token Ring wykorzystywała prototyp sieci zbudowanej w laboratorium IBM w Zurychu. Jak widać komitet doradczy IEEE miał do zaopiniowania aż trzy typy sieci lokalnych:

- system sieci wykorzystywany i proponowany przez firmy DEC, Xerox, Intel i ludzi z kręgu automatyzacji biur,
- system lokalnej sieci zbudowany przez firmę GM, jej podwykonawców i dostawców,
- system utworzony przez firmę IBM.

[40] Wg [TAN89].

W końcu zaaprobowano trzy standardy sieci lokalnych: *IEEE 802.3* (wykorzystujący *Ethernet*), *IEEE 802.4* (wykorzystujący *Token bus*) oraz *IEEE 802.5* (wykorzystujący *Token Ring*).

Firma GM i wiele przedsiębiorstw przemysłowych zainteresowanych automatyzacją procesów przemysłowych w celu uniknięcia problemów niekompatybilności zauważyli potrzebę wprowadzenia protokołów komunikacyjnych do warstw modelu OSI, co zaowocowało utworzeniem sieci MAP (*Manufacturing Automation Protocol*) stosowanej w świecie przemysłu.

W tym samym czasie firma Boeing zainteresowana była wprowadzaniem standardów w świecie automatyzacji prac biurowych. Ze względu na cechy sieci Ethernet zdecydowano się na wykorzystanie protokołów sieci Ethernet, co zaowocowało utworzeniem sieci TOP (*Technical and Office Protocol*). Chociaż sieć MAP i TOP różniły się od siebie ze względu na najniższe warstwy modelu OSI, GM i Boeing pracowały wspólnie nad utworzeniem protokołów zapewniających pełną kompatybilność.

USENET

W momencie wprowadzenia systemu operacyjnego UNIX pojawiła się konieczność kopiowania zbiorów między dwoma systemami operacyjnymi UNIX. W odpowiedzi pojawił się program *uucp* (ang. *Unix to Unix CoPy*). Dało to początek tworzeniu się nieformalnej sieci komputerów, w której dana maszyna automatycznie łączyła się z inną w celu przesyłania danych i poczty elektronicznej. Sieć rozwijała się dość szybko, gdyż jedynymi elementami koniecznymi do przyłączenia się do niej były: komputer pracujący w systemie operacyjnym UNIX oraz modem. Wspomniane elementy posiadała większość zachodnich uniwersytetów. Z czasem więc sieć bazująca na programie uucp rozrosła się do ponad 10 000 komputerów i ponad miliona użytkowników i określana była jako *UUCP*.

Sieć UUCP nie była zarządzana centralistycznie w przeciwieństwie do wielu sieci publicznych, czy sieci ARPANET. Europejska część nosiła nazwę *EUnet* i posiadała bardziej zorganizowaną strukturę. Każdy kraj w Europie posiadał odpowiednią bramę (ang. *gateway*) zarządzaną przez administratora. Administratorzy byli w ścisłym kontakcie między sobą kontrolując rozwój sieci.

Jedynym serwisem oferowanym przez sieć UUCP była poczta elektroniczna. Z czasem jednak część sieci UUCP zaczęto określać jako *USENET*. Stało się tak głównie dzięki inicjatywie podjętej przez *Uniwersity Duke* oraz *University of North Carolina*, gdzie zaoferowano usługę znaną jako grupy dyskusyjne (ang. *network news*).

Obecnie w sieci USENET istnieje wiele rodzajów grup dyskusyjnych (języki programowania, tematy humanistyczne, systemy operacyjne, mikrokomputery, ogłoszenia dotyczące pracy itd.). Użytkownik może wysłać wiadomość (przy wykorzystaniu programu *uucp*), która jest zwykle kopiowana do wszystkich użytkowników sieci USENET należących do danej grupy.

CSNET

Do 1980 r. powszechnie znana była wartość sieci ARPANET, jednak głównym problemem był fakt, że była ona w posiadaniu departamentu obrony i tylko uniwersytety i instytucje mające odpowiednią umowę mogły z niej korzystać. Aby rozległa sieć komputerowa mogła służyć szerokiemu gronu naukowców i studentów, NSF (ang. *National Science Foundation*) powołał do istnienia sieć *CSNET*, która była dostępna dla każdego wydziału informatyki na terenie Stanów Zjednoczonych. Z czasem CSNET stała się megasiecią, wykorzystując możliwości transmisji danych oferowane przez inne sieci komputerowe oraz dodając protokół tworzący ze zbioru sieci jedną logiczną sieć.

BITNET

Sieć BITNET (ang. *Because It's Time NETwork*) zarządzana przez organizację CREN (ang. *Corporation for Research and Educational Networking*), powstała w 1981 r. w City University of New York oraz Yale University. Twórcom sieci BITNET przyświecała idea stworzenia uniwersyteckiej sieci komputerowej jednak w odróżnieniu od sieci CSNET, sieć Bitnet miałaby łączyć wszystkie wydziały uniwersyteckie, a nie tylko informatyczne[41].

Według danych z 1989 r. sieć ta miała 175 komputerów bazowych w Stanach Zjednoczonych i ponad 260 w Europie, gdzie do 1993 r. określana była jako sieć EARN (European Academic Research Network). Obecnie w sieci tej jest około 2500 komputerów bazowych zainstalowanych głównie w ośrodkach uniwersyteckich: w Stanach Zjednoczonych, w Kanadzie, w Meksyku, w Ameryce Południowej, w Europie i w Japonii.

Każdy komputer bazowy w sieci BITNET posiada połączenie z innym komputerem bazowym. W odróżnieniu od innych sieci[42] komputery bazowe same pełnią rolę serwerów komunikacyjnych.

SNA

SNA (ang. *System Network Architecture*) jest preferowaną przez firmę IBM architekturą sieciową. Warto wspomnieć, że model ISO/OSI został zaprojektowany po utworzeniu systemu SNA, uwzględniając niektóre elementy systemu SNA[43].

Celem architektury sieciowej SNA było umożliwienie klientom IBM konstrukcje ich własnej sieci. Jeszcze przed zaprojektowaniem systemu SNA firma IBM dysponowała wieloma produktami komunikacyjnymi wykorzystującymi wiele standardów komunikacyjnych. Ideą systemu SNA była eliminacja istniejącego chaosu komunikacyjnego i utworzenie w miarę koherentnego systemu pozwalającego na łączenie w całość wielu komponentów sieciowych. Jednakże SNA

[41] Wg [TAN89], [SHE95].

[42] Np. CYPRES za [TAN89].

[43] Jak np. ideę warstw i ich funkcje.

jest systemem dość skomplikowanym. Ponadto SNA wykonuje dużą liczbę funkcji niewykonywalnych w innych systemach sieciowych.

Bulletin Board System (BBS)

Odmienną koncepcją pracy z komputerami są systemy typu *host*. Systemy tego typu pozwalają na samodzielną pracę i dostęp do serwera realizującego funkcje zarządzania plikami, drukarkami oraz systemem zarządzania danych. Do takich systemów należą systemy komputerowe typu BBS (ang. *Bulletin Board System*). Połączenie BBS uzyskuje się zazwyczaj za pośrednictwem łącza telefonicznego. Użytkownicy rejestrują się w systemie, aby odczytać lub wysłać wiadomość lub pliki.

BBS-y oferują systemy wymiany wiadomości, umożliwiające prowadzenie i obsługiwanie elektronicznej konwersacji. Wiadomości pozostawiane są na BBS-ie a rejestrujący się użytkownicy mogą je przeczytać. Niektóre systemy pozwalają na jednoczesną rejestracje wielu użytkowników. Dzięki czemu mogą oni prowadzić „konwersację na żywo". Zaawansowane systemy BBS pozwalają na założenie sieci BBS-ów pozwalających na łączenie się i wymianę informacji w ustalonych godzinach z drugim systemem BBS lub jednoczesne łączenie się kilku BBS-ów z centralnym systemem BBS pełniącym rolę *huba*.

Do zasadniczych cech oprogramowania obsługującego system BBS zalicza się[44]:

- możliwość obsługi wielu użytkowników (jednocześnie),
- możliwość pracy z innymi sieciami[45] lub innymi BBS-ami,
- system zabezpieczeń uniemożliwiający wtargnięcie do systemu nieautoryzowanym osobom oraz ochrona zasobów przed nieautoryzowanymi użytkownikami.

2.5. SIEĆ INTERNET

Globalna sieć Internet należy obecnie do największych sieci komputerowych na świecie. Oferuje coraz większą liczbę i zakres usług. Nie jest również bez znaczenia fakt dostępu do innych sieci komputerowych[46] oraz stale wzrastająca liczba profesjonalnych, komercyjnych serwisów informacyjnych dostępnych przez Internet[47].

Internet obejmuje kilkadziesiąt tysięcy sieci, ponad milion komputerów bazowych i zapewnia wymianę poczty elektronicznej między 50 milionami ludzi[48]. Porównuje się Internet do usług sieci CompuServe, Prodigy lub BIX (ang. *Byte Infor-*

[44] Wg [SHE95].
[45] Np. z sieciami FidoNet lub PCRelay – za [SHE95].
[46] Np.: X-25, BITNET.
[47] Np. Dialog-Datastar (dialog.com).
[48] Por.: [ENG95], [LIU94], [MAT95].

mation Exchange). Jednak CompuServe i inne sieci (np. sieć BBS – Bulletin Board System) są sieciami o małym zasięgu oferującymi usługi o wąskim zakresie, a ich aplikacje nie wykorzystują architektury klient-serwer. Internet może być porównywany do szkieletu komunikacyjnego, pozwalającego na dostęp do wielu usług, stanowiącego strukturę powiązaną z wieloma sieciami publicznymi i prywatnymi.

Początkiem sieci Internet był projekt ARPANET (por. podrozdz. 2.4.). W 1983 r. militarna część ARPANET została wydzielona jako MILNET (*Military Network*) pozostawiając możliwość wzajemnej komunikacji. W 1990 sieć ARPANET oficjalnie zakończyła swoje działanie[49].

Do określania wybranych zasobów (zbiorów) w sieci Internet coraz powszechniej stosuje się tzw. URL (Uniform Resource Locator):

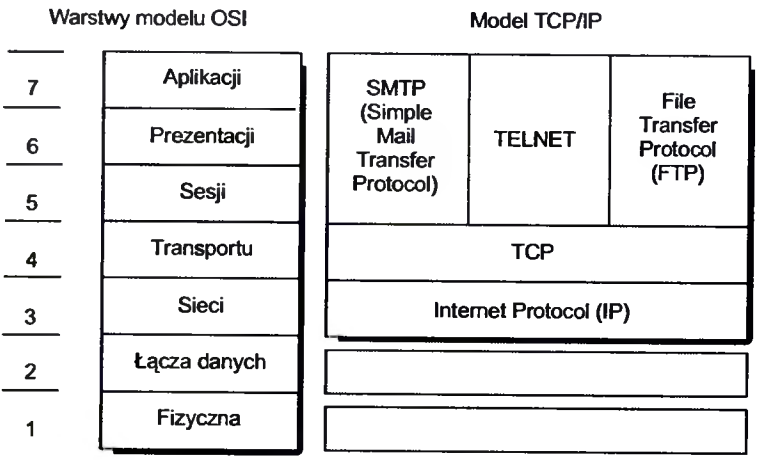
rodzaj serwera://adres.serwera/lokalizacja.zbioru/nazwa zbioru.

np.: *http://venus.ci.uw.edu.pl/uw/ibin/sss4.html*

Powyższy sposób oznaczania zbiorów będzie również przyjęty w niniejszej pracy.

Poniżej zostanie krótko omówiony jeden z najistotniejszych elementów sieci Internet – protokoły komunikacyjne TCP/IP.

Komunikacja w sieci Internet – protokoły TCP/IP



Rysunek 2.9 Porównanie modelu OSI i protokołów TCP/IP.

TCP/IP jest obecnie powszechnie obowiązującym standardem w światowej sieci Internet. W latach siedemdziesiątych w USA rozpoczęto badania nad znalezieniem sposobu, który pozwoliłby różnym typom komputerów używanych przez instytucje rządowe i uniwersytety na wzajemną komunikację. Problem nie był prosty, gdyż komputery te stosowały różne systemy operacyjne i oprogramowanie komunikacyjne. W końcu utworzono zbiór standardów pozwalających na łączenie różnego

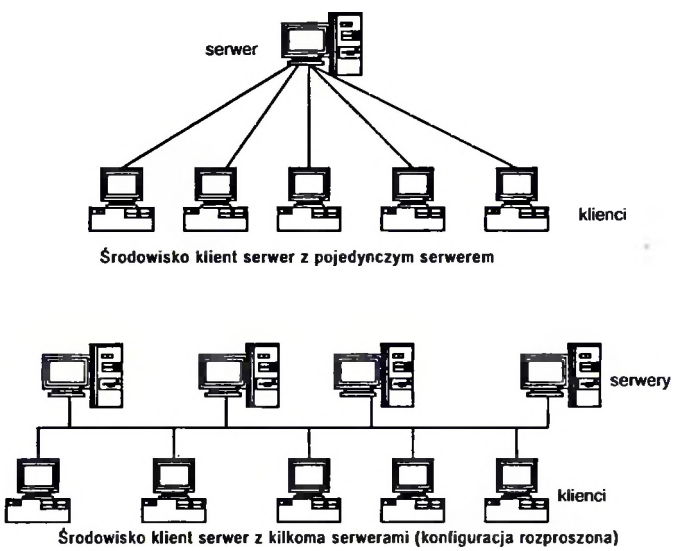
[49] Wg [SHE95].

typu komputerów i sprzętu komunikacyjnego oraz zbiór programów dających możliwość emulacji terminali, przesyłania plików pomiędzy różnymi systemami, czy zdalne wykonywanie programów. Zbiór tych standardów nazwany został *Transmission Control Protocol/Internet Protocol* (TCP/IP). TCP/IP wykorzystuje tzw. komunikację warstwową (por. model ISO/OSI), gdzie najniższe warstwy odpowiedzialne są za wspólną metodę komunikacji pomiędzy różnymi komputerami a sprzętem komunikacyjnym, warstwa środkowa określa „metody kierowania sygnału w odpowiednie miejsce", zaś warstwy najwyższe zapewniają „wspólne usługi sieciowe" (emulacja terminala, przesyłanie plików, zdalną rejestrację w systemie itp.). Powyższy rysunek przedstawia porównanie omówionego wcześniej modelu ISO/OSI oraz protokołów TCP/IP.

Architektura klient-serwer

Podstawą działania większości systemów informacyjnych w sieci Internet jest architektura klient-serwer. Architektura tego typu określa relację między stacją roboczą użytkownika (klientem, ang. *front-end*) a serwerem (ang. *back-end*). Funkcję klienta pełni przeważnie inteligentny system, dysponujący odpowiednio dużą mocą obliczeniową pozwalającą na przejęcie części obciążenia całego systemu. Interakcja między klientem a serwerem składa się z serii żądań i odpowiedzi.

Wymienia się kilka typów konfiguracji środowiska, niektóre z nich przedstawia rysunek 2.10[50].

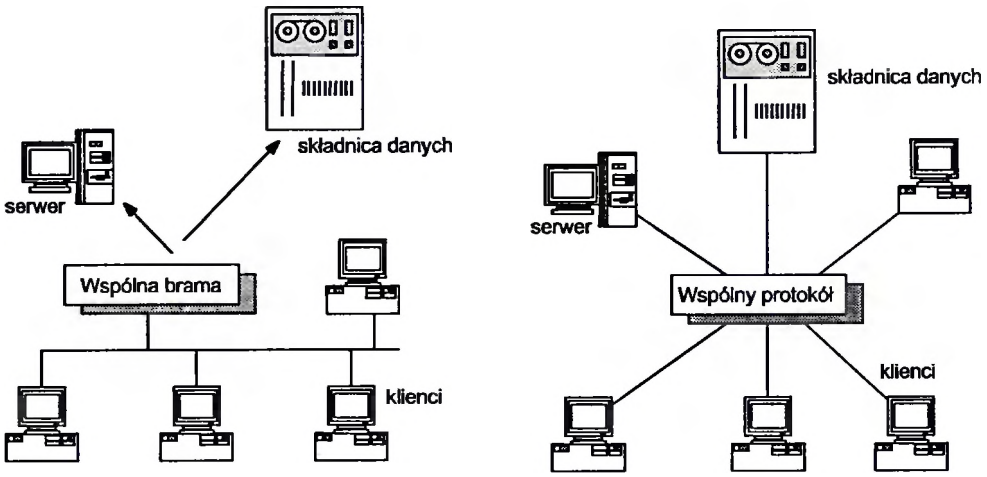


Rys. 2.10 Przykłady konfiguracji kient-serwer.

W sieciach typu partnerskiego (ang. *peer-to-peer*) stacje robocze mogą pełnić zarówno rolę klientów jak i serwerów. Użytkownik jest w stanie udostępniać zaso-

[50] Por. [SHE95].

by (pliki) ze swojego dysku twardego innym użytkownikom sieci. Oznacza to, że dana stacja może pełnić zarówno rolę serwera jak i klienta.



Rys. 2.11 Modele środowisk typu klient-serwer wykorzystujących bramę oraz wspólny protokół[51].

W przypadku sieci korporacyjnych zbudowanych na bazie wcześniej zainstalowanych sieci LAN, w których system klienta nie jest programowo zgodny z systemem serwera stosuje się dwa modele korzystania z danych korporacyjnych (patrz rysunek 2.11):

- zastosowania wspólnej bramy (gateway) przekształcającej żądania klientów do postaci zrozumiałej przez niekompatybilne programowo serwery,
- zastosowanie wspólnej warstwy protokolarnej, pełniącej rolę interfejsu, wykorzystującego jeden standardowy protokół, który pośredniczy między niekompatybilnymi klientami i serwerami.

Funkcje systemu klienta i serwera rozbite są na kilka procesów. W różnych systemach operacyjnych i aplikacjach zakres i liczba zadań realizowanych przez serwer są różne. W niektórych rozwiązaniach serwer wykonuje jak najmniej czynności, co zapewnia optymalne wykorzystanie jego mocy obliczeniowych przy pracy z dużą liczbą klientów.

W przeważającej liczbie konfiguracji komunikacja realizowana jest za pośrednictwem sieci lokalnych. Serwery mogą być geograficznie odległe od użytkownika, a dostęp do nich może być możliwy za pośrednictwem łącz telekomunikacyjnych (*serwery korporacyjne*). Serwery mogą również znajdować się na odległych miejscach dostępnych dla użytkowników za pośrednictwem łącz komunikacyjnych.

Do podstawowych korzyści wynikłych ze stosowania architektury typu klient-serwer zalicza się:

[51] Wg [SHE95].

- równomierne rozłożenie obciążenia zadaniami wynikłego z obsługi oprogramowania aplikacyjnego,
- zmniejszenie ilości informacji przesyłanych w sieci,
- zwiększanie bezpieczeństwa danych dzięki scentralizowanemu systemowi kontroli,
- zwiększenie szybkości przetwarzania danych dzięki możliwości równoległego i częściowego przetwarzania danych przez poszczególne komputery.

Dostęp do zasobów sieci Internet

Poniżej zostaną krótko omówione ogólne zasady korzystania z większości aplikacji dostępnych w sieci INTERNET na przykładzie systemu ARCHIE[52]. System ten służy do przeszukiwania adresów i katalogów komputerów bazowych dostępnych jako anonymous FTP[53]. Podobnie jak wiele innych aplikacji sieciowych osiągalny jest za pośrednictwem: lokalnego klienta, zdalnego klienta oraz przez pocztę elektroniczną.

```
archie>
#'search'(typestring)hasthevalue'sub'.


archie>findbolbook
#Searchtype:sub.
#Yourqueueposition:7
#Estimatedtimeforcompletion:2minutes,31seconds.
#working...-0+0
   (...)


 Hostftp.luth.se(130.240.18.2)
 Lastupdated23:04160ct1994

       Location/pub/msdos/win3/
         DIRECTORY drwxr-xr-x 1536bytes 07:07 17Aug1994 toolbook


  Hostgoliat.eik.bme.hu(152.66.115.2)
  Lastupdated04:02 20ct1994

        Location/pub/win3/
           DIRECTORY drwxrxr-x 512 bytes 16:49 11Aug1994 toolbook

     (...)
```

Rys. 2.12 Poszukiwanie przez odległego ARCHIE-klienta

„Zdalny klient" (ang. *remote client*) oznacza otwarcie sesji Telnet z danym komputerem bazowym (zawierającym program serwer-ARCHIE). Użytkownik

[52] Por. lokalnego klienta Archie (omówionego w dalszej części tego rozdziału) oraz zdalnego klienta Archie (rys. 2.12).

[53] System Archie oraz FTP zostaną szczegółowo omówione niżej.

rejestruje się w systemie odległego komputera[54] i po wydaniu odpowiedniego polecenia wyszukiwawczego otrzymuje po jakimś czasie rezultat wyszukiwania (por. rys. 2.12).

W przypadku poczty elektronicznej użytkownik wysyła list do najbliższego serwera, którego treścią jest odpowiednie polecenie wyszukiwawcze np. dla serwera ARCHIE: *find ToolBook*. Polecenie *find* oznacza, że system będzie dokonywać przeszukiwania dla każdej interpretacji podanego wzoru, co w podanym przypadku (*ToolBook*) może oznaczać nazwę programu, katalogu itp. Po jakimś czasie serwer ARCHIE wysyła do użytkownika „odpowiedź", którego treścią jest rezultat zleconego mu przeszukiwania.

Obecnie prawie wszystkie ważniejsze usługi sieci Internet dostępne są przez pocztę elektroniczną. Ze względu na charakter poczty elektronicznej eliminuje to jednak możliwość korzystania z bogatego interfejsu użytkownika i wielu dodatkowych elementów, jednak w wielu wypadkach jest jedyną drogą uzyskania informacji.

2.6. USŁUGI INFORMACYJNE W SIECI INTERNET

Poniżej przedstawione zostaną wybrane usługi sieci Internet[55].

Do podstawowych usług sieci Internet można zaliczyć: pocztę elektroniczną, FTP oraz Telnet.

Poczta elektroniczna

Poczta elektroniczna (e-mail) jest najczęściej wykorzystywanym sposobem przekazywania informacji. Warunkiem korzystania z poczty elektronicznej jest posiadanie dostępu do serwera obsługującego pocztę elektroniczną oraz odpowiedni program klienta (rys. 2.13).

Treścią listu winien być prosty tekst w kodach ASCII zawierający nie więcej niż 80 znaków w wierszu. Możliwe jest zatem przesyłanie pocztą np. dokumentów sformatowanych w TEX'u. Dzięki możliwości kodowania zbiorów binarnych na podzbiór znaków ASCII[56] istnieje również możliwość przesyłania pocztą elektroniczną plików binarnych[57].

Istotnym elementem jest mechanizm dostarczania poczty – protokół SMTP[58] (ang. *Simple Mail Transfer Protocol*). SMTP[59] daje możliwość dostępu do poczty

[54] Np. Telnet 130.240.12.30.

[55] Podział wykorzystuje schemat przyjęty w opracowaniach EARN (European Academic Research Network).

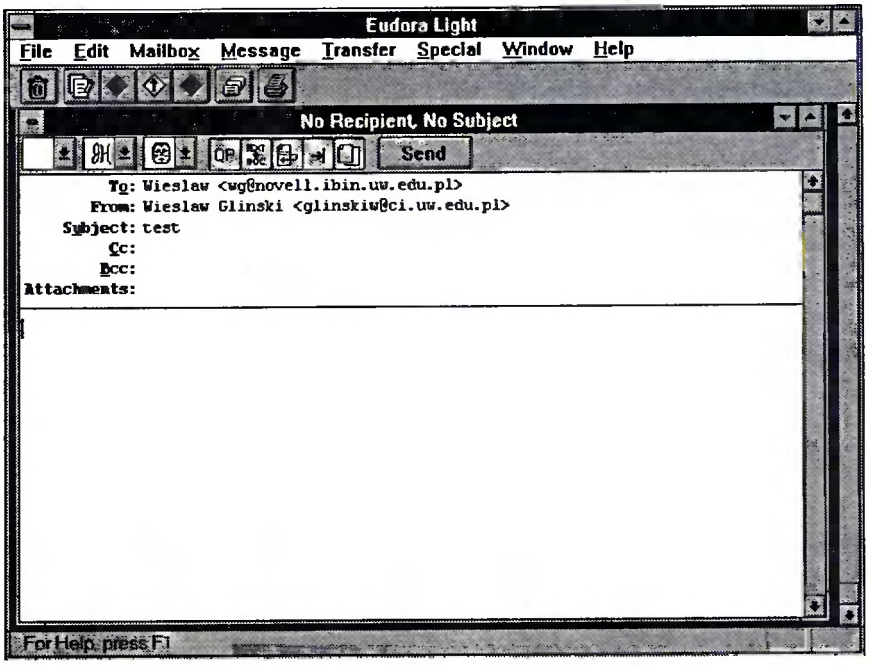
[56] Np.:uuencode, uudecode.

[57] Za [RYK94].

[58] Por. modele TCP/IP i OSI.

[59] Por. ISO/OSI i TCP/IP.

w sieci Internet z każdego systemu, który zapewnia ten protokół. Poczta realizowana za pomocą SMTP pozwala użytkownikom komputerów PC na wymianę korespondencji z użytkownikami systemów UNIX czy VM korzystających z Internetu bez pośrednictwa specjalnych bram (ang. *gateway*).



Rys. 2.13 Program Eudora – klient obsługi poczty elektronicznej

FTP

FTP (File Transfer Protocol) daje możliwość przesyłania plików pomiędzy komputerami (patrz rys. 2.14).

W tym miejscu należy zaznaczyć, że istnieje duża grupa niekomercyjnych programów (typu: *public domain, shareware* czy *freeware*) dostępnych w sieci Internet. Dostęp do nich możliwy jest właśnie przez wspomniany FTP. Użytkownik powinien znać adres komputera bazowego, na którym ulokowane są poszukiwane zasoby, ale i tu mogą mu przyjść z pomocą wspomniana wyżej aplikacja w sieci Internet-ARCHIE – tj. system wyszukujący adresy komputerów bazowych w sieci Internet. Warto zaznaczyć, że oprogramowanie (shareware, public domain itp.) dostępne jest za pośrednictwem tzw. *anonymous FTP*[60]. Rysunek 2.14 przedstawia przykładową sesję FTP z komputerem bazowym – novell.ibin.uw.edu.pl[61].

[60] Oznacza to, że po uzyskaniu połączenia z danym komputerem należy wpisać w miejsce pytania o nazwę użytkownika (ang. *login name:*) *anonymous*, a w miejsce hasła, swój adres e-mail w sieci Internet.

[61] Jest to również serwer sieci lokalnej IBIN UW dostępny w sieci Internet (por. rozdział 5).

```
ftp 148.81.213.2
VM TCP/IP FTP V2R1
Connecting to 148.81.213.2, port 21
220-It is IBIN FTP Server at Warsaw University
220 FTP Server for NW 3.11, 400 (v1.9), (c) 1993 HellSoft
anonymous
>>>USER anonymous
331 Anonymous Login OK, send id as password
Password
>>>PASS ********
230-User Logged.
230-You are anonymous FTP user connected to IBIN Server at
Warsaw Uniwersity
230 Home Directory:/SYS/PUB
Command:
```

Rys. 2.14 Przykładowa sesja FTP

Telnet

Podstawową funkcją programu Telnet jest umożliwienie użytkownikowi rejestrację na odległych komputerach bazowych. Początkowo program Telnet był prostym programem emulacji terminala, wysyłającym całą informację wprowadzoną przez użytkownika do odległego komputera bazowego (host), z czasem jednak powstawały nowsze wersje realizujące lokalnie część procesu przetwarzania[62].

```
VM/XA ONLINE
              IBM            UNIWERSYTET*WARSZ             IBM
                    UNIWERSYTET*WARSZAWSKI*UNIWERSY
                UNIWERSYTET*WARSZAWSKI*UNIWERSYTET*WARSZA
              WARSZAWSKI*UN                      WARSZAWSKI*U
             UNIWERSYTET*W        IIIIIII         UNIWERSYTET
             WARSZAWSKI*U         IIIIIII
             UNIWERSYTET
             WARSZAWSKI*        IIIIIIIII
             UNIWERSYTET           IIIIIII
             UNIWERSYTET*         CENTRUM
             UNIWERSYTET*W      INFORMATYCZNE      UNIWERSYTET
               WARSZAWSKI*UNI                     WARSZAWSKI*U
                 UNIWERSYTET*WARSZAWSKI*UNIWERSYTET*WARSZA
                      UNIWERSYTET*WARSZAWSKI*UNIWERSYT
              IBM            UNIWERSYTET*WARSZ             IBM

Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID   --->
PASSWORD --->

COMMAND  --->
```

Rys. 2.15 Przykładowa sesja Telnet

Jak wszystkie przedstawione tutaj programy sieci Internet, jest on systemem typu klient-serwer, w którym użytkownik uruchamia aplikację Telnet i zestawia

[62] Wg [SHE95]s.1045.

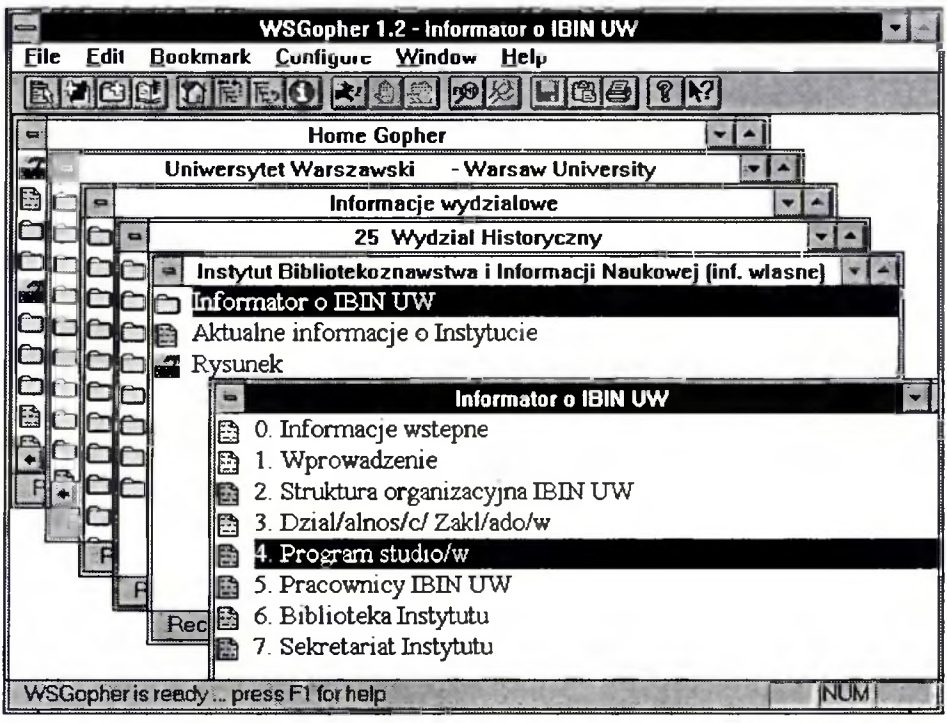
łącze do procesu Telnet uruchamianego na odległym komputerze. Polecenia wprowadzane są z klawiatury i przyjmowane przez część klienta programu Telnet. Następnie Telnet-klient przesyła odpowiednie żądania do serwera programu Telnet na odległym komputerze bazowym. Dzięki temu użytkownik może uruchamiać niektóre programy dostępne na odległym komputerze i pracować z nimi, jak gdyby był bezpośrednio podłączony do odległego komputera bazowego.

Systemy informacyjne w sieci Internet

Ostatnio coraz większą popularność zyskały sobie systemy informacyjne sieci Internet o charakterze hipertekstowym (por. rys. 2.18). Informacja w systemach Gopher i WWW nie ma charakteru typowej bazy danych i nie jest tym samym dobrze ustrukturalizowana (pola, podpola, rekordy itp.). Poniżej zostaną omówione: Gopher, Veronica oraz najpopularniejszy obecnie World Wide Web (WWW).

System Gopher

System Gopher jest rozproszonym systemem informacyjnym. Oznacza to, że użytkownik widzi zbiór dokumentów znajdujących się na różnych komputerach bazowych jako jeden. Istnieje możliwość przechodzenia do poszczególnych części dokumentu bez znajomości położenia danych informacji i sposobu dotarcia do nich. Informacja w systemie Gopher przedstawia się użytkownikowi jako zbiór za-



Rys. 2.16 Informacja o IBIN UW dostępna w systemie Gopher

gnieżdżanych menu. System ten przypomina strukturę katalogów i zbiorów w systemach Unix czy MSDOS. Poszczególne menu mogą być ulokowane na lokalnym komputerze lub na innych komputerach bazowych obsługiwanych przez inne serwery systemu Gopher. W skład systemu Gopher mogą wchodzić zbiory binarne lub tekstowe. Ponadto Gopher daje możliwość dostępu do innych systemów informacyjnych (World-Wide-Web, WAIS, Archie, WHOIS) i usług sieci Internet (Telnet, FTP).

Z oprogramowania Gopher można korzystać na trzy sposoby: wykorzystując oprogramowanie lokalnego Gopher-klienta (por. rysunek 2.16), przez pocztę elektroniczną[63] oraz wykorzystując oprogramowanie odległego Gopher-klienta[64]. Rys. 2.16 przedstawia system Gopher klient dla środowiska Windows.

Veronica

System Veronica był próbą rozwiązania problemu nawigacji w sieci serwerów Gopher (ang. *Gopherspace*) dając możliwość wyszukiwania odpowiednich menu systemów Gopher. Veronica pozwala na odnajdywanie informacji w systemie Gopher bez przechodzenia przez szereg menu Gophera. System ten pełni podobną rolę jak system Archie dla anonymous FTP.

Przeważnie system Veronica jest dostępny z menu systemu Gopher[65], pozwalając na wyszukiwanie menu serwerów Gopher według podanego wzoru. System daje również możliwość zastosowania operatorów boolowskich (*and, or, not*) oraz znaków globalnych (*).

Np. wprowadzając wyrażenie wyszukiwawcze :

Eudora and Macintosh

system odpowiada podaniem odpowiedniego menu zawierającego zarówno słowa: *Eudora* jak i *macintosh* (por. rysunek 2.17).

```
Eudora: Popmail for the Macintosh.
v4.1 EUDORA:  E-MAIL  FOR THE MACINTOSH
Micro News: EUDORA - A Mailer for the Macintosh
Eudora: Electronic Mail on Your Macintosh.
ACS News - Eudora Mail Reader for Macintosh
    (...)
```

Rys. 17 Wyszukiwanie w systemie Veronica[66]

Dostępność systemu Veronica możliwa jest również przez pocztę elektroniczną oraz zdalnego klienta[67].

---

[63] W treści listu do jednego z serwerów (gophermail@calvin.edu, gopher@earn.net, gopher@dsv.su.se, gomail@ncc.go.jp). konieczne jest podanie dokładnej lokalizacji menu (lub części) systemu Gopher. (s. 7 [EAR95]).

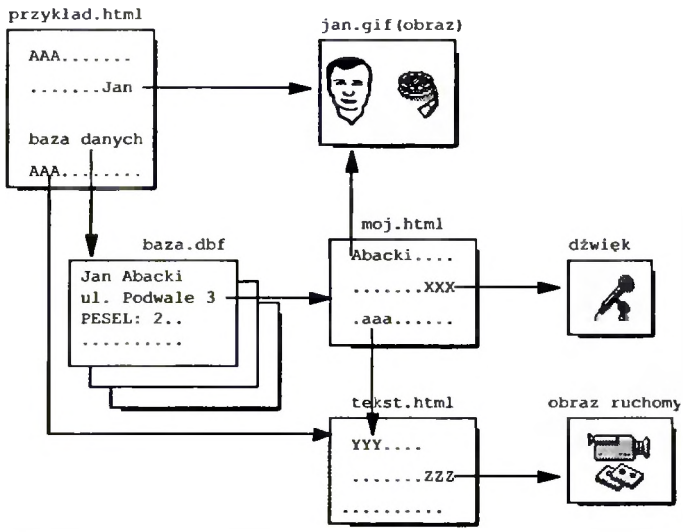
[64] Np. odległy klient na komputerze bazowym: gopher.chalmers.se [EAR95].

[65] Jako (ang.) *Other Gopher servers...*

[66] Wg [EAR93].

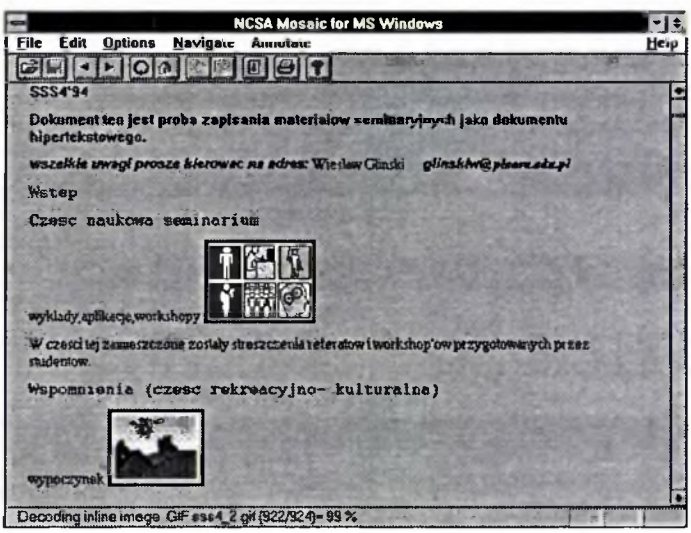
[67] Wg [EAR93], [EAR95].

WWW (World-Wide Web)



Rys. 2.18 Przykłady połączeń w dokumentach hipermedialnych WWW.

Zasoby informacyjne WWW odgrywają najważniejszą rolę w omówionym w rozdz. 4 i 5 systemie NetExp. System WWW[68] jest rozproszonym systemem informacji hipertekstowej (hipermedialnej), pozwalając użytkownikowi na przechodzenie do poszczególnych dokumentów (tekstowych, graficznych, zbiorów baz danych itp.). Rysunek 2.18 obrazuje ideę systemów hipermedialnych.



Rys. 2.19 Przykładowa sesja z serwerem WWW przez Netscape

[68] WWW określany jest też jako W3.

Netscape, Mosaic są typowymi klientami systemów WWW. Rysunek 2.19 przedstawia przykładową sesję z serwerem WWW przez Netscape.

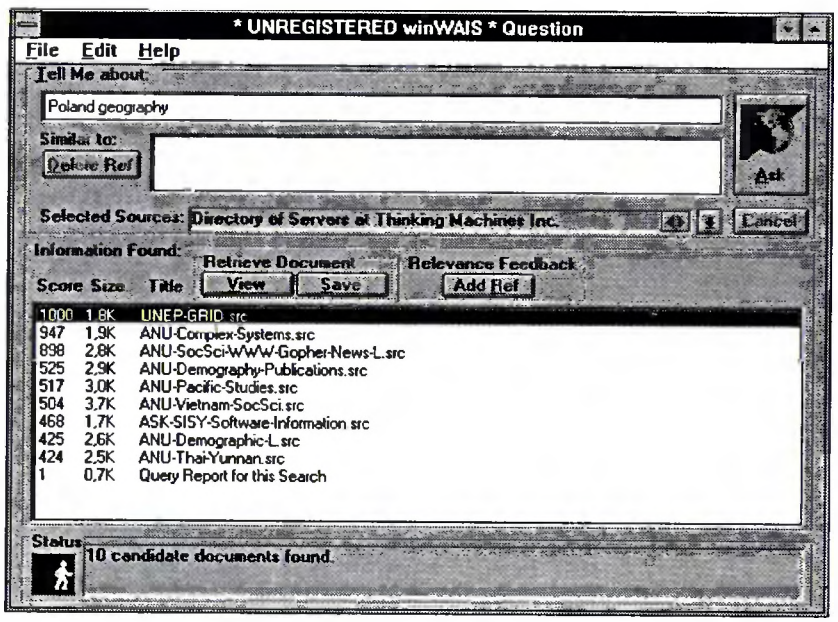
Dokumenty hipermedialne połączone są ze sobą przez odpowiednie grupy wyrazów (zbiorów graficznych itp.). Gdy np. przedstawiane jest nowe słowo w danym dokumencie, system hipertekstowy umożliwia użytkownikowi przejście do innego dokumentu związanego z danym słowem (np. wyjaśniającym jego znaczenia).

Podobnie jak w systemie Gopher (który ma strukturę hierarchiczną) użytkownik nie musi wiedzieć, gdzie znajduje się dany dokument lub jego część. Z matematycznego punktu widzenia dokumenty hipertekstowe mają postać grafu nieskierowanego. Większość dokumentów hipertekstowych dla serwerów WWW zapisanych jest w języku HTML (*HyperText Murkup Language*)[69].

Chociaż najlepiej WWW prezentuje się w przypadku lokalnego klienta, to istnieje możliwość dostępu do niego przez pocztę elektroniczną[70] oraz zdalnego klienta[71].

Przeszukiwanie baz danych

System WAIS



Rys. 2.20 WAIS klient dla środowiska Windows

[69] Dokument hipertekstowy zapisany w standarcie HTML jest zbiorem tekstowym kodowanym według normy ISO-8859; PN-93 T-42118 ([MAC95]).

[70] Np.: serwer-listserv@www0.cern.ch; w treści listu komenda send *adres URL* [EAR95].

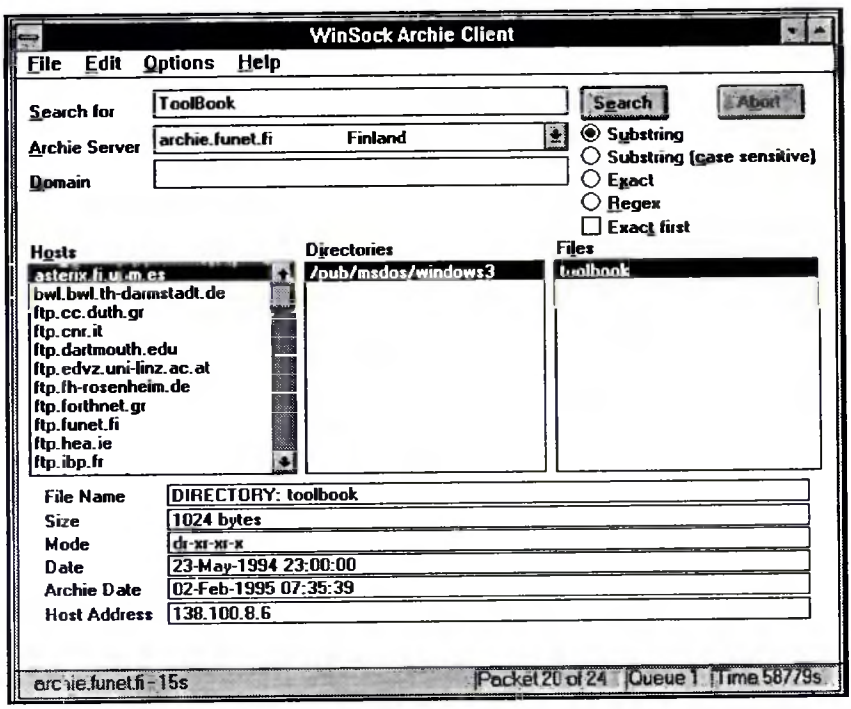
[71] Np.: serwer-info.funet.fi; *login name: www* [EAR95].

Wide Area Information System (rysunek 2.20) wydaje się szczególnie istotny dla informacji naukowej. Jest dość ważnym narzędziem do wyszukiwania poindeksowanych dokumentów[72]. W systemie WAIS prawie każde słowo jest indeksowane, do opisu dokumentów stosuje się standard ANSI Z39.50, ponadto każda odpowiedź posiada współczynnik dokładności (od 1 do 1000). WAIS pozwala wyszukiwać w archiwach sieci Internet dokumenty zawierające określone grupy słów. Jak większość systemów sieci Internet wykorzystuje on architekturę typu klient-serwer. Rysunek 2.20 przedstawia wersję klienta-WAIS dla środowiska Windows.

Poszukiwanie zasobów sieci

ARCHIE

System Archie[73] stworzony przez Alana Emtage, Petera Deutcha i Billa Heelana z Uniwersytetu McGill w Kanadzie (McGill University Computing Center)[74] jest systemem oferującym katalog zasobów w sieci Internet. Najbardziej znanym zastosowaniem systemu Archie jest wyszukiwanie zasobów dostępnych przez tzw. *anonymous ftp*.



Rys. 2.21 Poszukiwanie przez ARCHIE-klienta

[72] Por. [KRO93].
[73] System ten został skrótowo przedstawiony przy okazji omówienia usługi FTP.
[74] Za [EAR93].

Przeciętnie raz na miesiąc serwer Archie uruchamia program, który automatycznie dokonuje przeglądu wszystkich zarejestrowanych serwerów FTP i generuje listę dostępnych zbiorów. Zbiory dostępne przez anonymous ftp są przeważnie programami użytkowymi dla systemów Windows, MSDOS, Macintosh, UNIX itd.). Informacja opisująca dany zbiór w systemie Archie określa nazwę komputera bazowego, (jego IP, i nazwę odwołującą się do DNS[75]), nazwę katalogu oraz nazwę zbioru. Rysunek 2.21 przedstawia sesję z serwerem Archie przez Archie-klienta[76].

HYTELNET

System Hytelnet Petera Scotta (*Systems Department, University of Saskatchewan Libraries*)[77] jest prostym systemem hipertekstowym zawierającym: adresy komputerów bazowych sieci Internet osiągalnych przez sesję Telnet (pozwalającą m.in. na dostęp do: zasobów bibliotek, serwerów Gopher, WAIS, systemów WWW itd.), informacje o korzystaniu z katalogów bibliotecznych online oraz słownik sieci Internet. Baza danych systemu Hytelnet jest przesyłana do komputera użytkownika i składowana lokalnie, jest zatem możliwe jej uaktualnianie i wprowadzanie zmian. Obecnie istnieje nowa wersja systemu Hytelnet dostępna w wersji HTML jako strona WWW[78]. Polecenia zawarte w systemie Hytelnet ułatwiają użytkownikowi łączenie się z wybranymi komputerami bazowymi[79].

Odnajdywanie osób i komputerów bazowych

```
Whois: PERSON GLINSKI

Glinski, Monica <MG705>              monica@FOM.COM       <313> 396-5761
Philip, Monica <PG296>             digallery@MSN.COM     716-832-2861

Whois: HOST KENT

[No name] <KENT>          KENT.ANSTO.GOV.AU  137.157.45.204
Kent State University <CONDOR1> NS.MCS.KENT.EDU  131.157.45.204
Kent State University <KENT-HST> ZEUS.KENT.EDU  131.123.75.254
Kent State University <KSUVXA> KSUVXA.KENT.EDU  131.123.1.1
```

Rys. 2.22 Wyszukiwanie w systemie WHOIS81 przez odległego klienta

[75] DNS – *Domain Name System* – System Domen Nazw Internetowych, dzięki czemu wiadomo, że. 148.81.213.2 oznacza to samo co *novell.ibin.uw.edu.pl*..

[76] Więcej informacji na temat systemu Archie Czytelnik znajdzie w następujących pracach: [CIES95], [EAR93a], [EAR93b] oraz pod adresami info@bunyip.com archie-group@bunyip.com oraz w liście dyskusyjnej archie-people@bunyip.com.

[77] Za [EAR93].

[78] Por. http://www.cc.ukans.edu/hytelnet_html/STAR.TXT.html oraz http://www.cc.ukans.edu/hytelnet_html.tar.Z.

[79] Więcej informacji można znaleźć pod adresem aa375@freenet.carleton.ca, artykuł Petera Scotta dostępny jest pod adresem listserv@uhupvm1.uh.edu (w treści listu ma być podana wiadomość *GET SCOTT PRV3SN4 F=MAIL*), lista dyskusyjna o systemie Hytelnet: HYTEL-L na serwerze list dyskusyjnych listserv@kentvm.kent.edu.

Systemy WHOIS[80] oraz NetFind pozwalają odnajdywać adresy poczty elektronicznej, adresy pocztowe, numery telefonów użytkowników sieci Internet (patrz rys. 2.22). Mogą również dostarczyć informacji o sieciach, organizacjach związanych z sieciami komputerowymi, informacje o komputerach bazowych sieci Internet. Obecnie poszczególne komputery bazowe sieci Internet starają się utrzymywać bazy danych dotyczące tylko swoich użytkowników.

Usługi związane z przesyłaniem zbiorów

Wśród usług związanych z przesyłaniem zbiorów można wymienić TRICKLE pozwalający na przesyłanie zbiorów dostępnych przez tzw. *anonymous FTP*. Pewną odmianą systemu TRICKLE dla użytkowników sieci BITNET jest program BITFTP. Zadanie przesłania określonego zbioru w określonym katalogu i na danym komputerze bazowym jest kierowane do serwera BITFTP, który wykonuje zlecone mu zadanie.

Biblioteki cyfrowe

Istotnym zagadnieniem z punktu widzenia sieci informacyjnych są biblioteki cyfrowe (ang. *digital libraries, electronic libraries*). Biblioteki cyfrowe definiuje się w różny sposób:

- system informacyjny składający się z danych tekstowych[82],
- zbiór rozproszonych usług informacyjnych[83],
- połączenia między rozproszoną informacją[84],
- sieciowy system informacji multimedialnej[85].

Omówiono bardzo pobieżnie tylko wybrane aplikacje w sieci Internet, część z nich jest stale udoskonalana, pozostałe przestają odgrywać istotną rolę i ulegają zapomnieniu. Jednak wobec coraz większego zainteresowania siecią Internet, jej rolą i rosnącymi możliwościami dostępu do niej powstaje coraz większa liczba programów wspomagających niedoświadczonego użytkownika w wyszukiwaniu informacji, do nich należą tzw. inteligentne systemy (*inteligent agent*).

2.7. INTELIGENTNE SYSTEMY W SIECI INTERNET

W omówieniach większości inteligentnych systemów wspomagających wyszukiwanie informacji w sieciach rozległych podkreśla się rolę tzw. inteligentnych agentów (IA), wspomagających użytkownika w wyszukiwaniu informacji. Wymienia się dwie istotne cechy agentów:

[80] Usługa WHOIS była pierwotnie określana jako *NICKNAME*([EAR95]).
[81] Wykorzystano zdalnego klienta: whois.internic.net (IP:198.41.0.6), userid:*whois*.
[82] Wg [CRO95].
[83] Wg [WIL95].
[84] Wg [SCH95].
[85] Wg [FAKL95].

- abstrakcja (ang. *abstraction*) oznacza, że wszelkie szczegóły technologiczne związane z systemem są niewidoczne dla użytkownika a IA określa, gdzie znajdują się relewantne zasoby,
- *dystrakcja* (ang. *distraction*) oznacza, że wszelkie złożone i uciążliwe dla użytkownika działania są podejmowane przez agenta,

Dla inteligentnych agentów ważniejszy jest proces tzw. abstrakcji niż dystrakcji. Chociaż obecnie staje się coraz bardziej modny styl interakcji użytkownika z systemem (kliknięcie myszą, przesuwanie ikon itp.), to przypuszcza się, że nie będzie on miał istotnego znaczenia w procesie samego wyszukiwania informacji. Według Alana Kaya[86] same narzędzia wyszukiwawcze nie odegrają istotnej roli, gdyż „nikt nie będzie chciał spędzać wiele godzin w poszukiwaniu setek sieci z miliardami potencjalnie istotnych informacji". I chyba należy się zgodzić ze stwierdzeniem, że proces ten powinien być powierzony inteligentnym agentom, którzy będą w stanie właściwie zinterpretować potrzeby użytkownika i samodzielnie dokonać wyszukania.

Rola inteligentnych „agentów"

Termin inteligentny agent jest określany na wiele sposobów. Przeważnie oznacza on program komputerowy zachowujący się w sposób analogiczny do pośrednika informacji (ang. *information agent, information broker*). Wymienia się następujące elementy inteligentnych agentów[87]:

⇒ Autonomia oznacza, że agent przejmuje kontrolę nad swoimi własnymi działaniami w następujący sposób:
- Działania kierowane celami – agent rejestruje tylko ogólnie określone cele wskazujące na żądania użytkownika, ponadto agent jest odpowiedzialny za sposób ich realizacji,
- Łatwość współpracy z użytkownikiem – agent nie wykonuje ślepo poleceń, ale jest w stanie je modyfikować, zadając użytkownikowi dodatkowe pytania wyjaśniające lub nawet w szczególnych przypadkach odmawiając wykonania niektórych poleceń,
- Łatwość zmian – działania agenta nie są z góry określone, ale są uzależnione od zmian otoczenia,
- Automatyczna inicjacja systemu – w przeciwieństwie do standardowych programów uruchamianych przez użytkownika, IA „śledzi" zmiany zachodzące w otoczeniu i w zależności od nich podejmuje działania.

⇒ Ciągłość działania – działanie agenta jest ciągłe, nie jest jednorazowym działaniem spowodowanym żądaniem użytkownika;

⇒ „Osobowość" – agent posiada przyjazne metody i narzędzia komunikacji z użytkownikiem;

[86] Wg [BGS95].
[87] Wg [BGS95], [http://www.cs.umbcedu/agents/].

⇒ Łatwość komunikacji – agent jest w stanie komunikować się z innymi inteligentnymi agentami (obiektami) w tym z ludźmi w celu osiągnięcia określonego(ych) celu(ów);

⇒ Zdolność do adaptacji – agent może automatycznie przystosować się do użytkowników na podstawie poprzednich doświadczeń, może również automatycznie adaptować się do zmian otoczenia;

⇒ Mobilność – *agent* jest niezależny od konkretnej platformy sprzętowo-programowej.

Chociaż nie istnieje system posiadający wszystkie wyżej wymienione cechy, to istniejące prototypy inteligentnych agentów posiadają już większość z podanych wyżej cech.

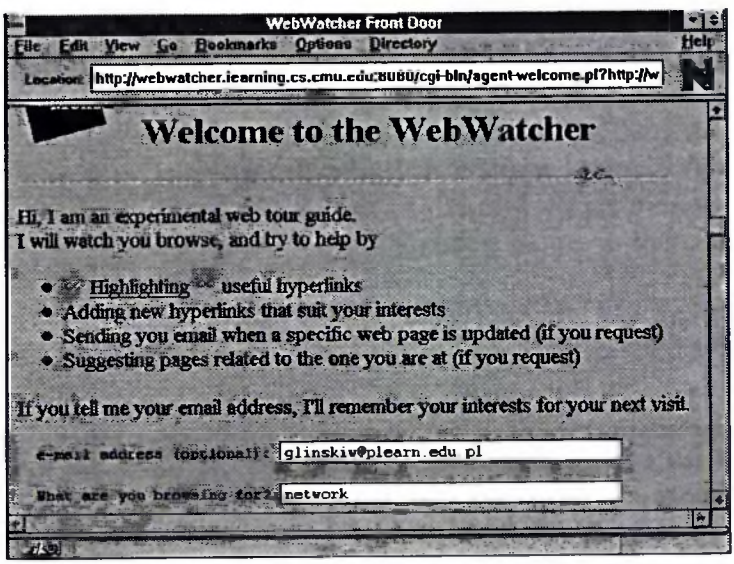
Typy agentów

Wyróżnia się kilka typów agentów w zależności od rodzaju zadań, jakie podejmują:

- Przewodnicy,
- Systemy indeksujące,
- Systemy „FAQ"

Przewodnicy

Jednym z inteligentnych agentów pełniącym rolę przewodnika jest WebWatcher (patrz rys. 2.23) pomagający użytkownikom poruszać się po światowej pajęczynie WWW. Np. system WebWatcher doradza użytkownikowi w wyborze kolejnego połączenia hipertekstowego. System uczy się przez obserwację zachowań użytkownika.



Rys. 2.23 System WebWatcher dostępny przez WWW

Systemy indeksujące

Kolejnym typem systemów (agentów) są systemy indeksujące. Do najbardziej znanych zalicza się: Lycos, WebCrawler i Infosek. Agenci indeksujący przeszukują znaczną liczbę dokumentów WWW tworząc indeksy dokumentów na podstawie ich tytułów oraz zawartości. Użytkownik może zadać pytanie polegające na wyszukaniu danego słowa(ów) kluczowego(ych) w tekście. Agenci indeksujący są w stanie dostarczyć szybkiej odpowiedzi, jednak nie zawsze jest ona adekwatna do postawionego pytania. Liczba fałszywych odpowiedzi wzrasta w miarę rozwoju WWW, poza tym IA nie są w stanie objąć swym zasięgiem informacji (utrzymywanych w bazach danych, serwerach np. WAIS i innych serwisach informacyjnych) nie będących częścią WWW.

Obecnie indeksujący agenci nie posiadają ograniczeń w tworzeniu indeksów, ich bazy są możliwie jak najbardziej wyczerpujące. Wedle przewidywań[88] za parę lat agenci indeksujący będą w stanie wybierać informację kierując się takimi czynnikami jak: przewidywane koszty dostępu, czy szybkości dostępu do danych. IA będą „filtrować" rezultaty poszukiwań na podstawie wskazówek dotyczących każdej strony WWW. Wersje wspomnianych inteligentnych agentów indeksujących są obecnie rozwijane przez Uniwersytet w Waszyngtonie (*University of Washington*)[89].

Systemy „FAQ"

Jednym z systemów bardziej selektywnych w doborze informacji są systemy pomagające użytkownikom odnaleźć odpowiedzi na najczęściej zadawane pytania znane pod skrótem FAQ (ang. *Frequently Asked Questions*), czyli tzw. agenci FAQ. Pomysł systemów typu „FAQ" wywodzi się z list dyskusyjnych i dotyczy najczęściej zadawanych przez użytkowników pytań i odpowiedzi na dane tematy (np.: nauki społeczne, polityka, ekonomia, informatyka itd.). Warto zaznaczyć, że zbiory typu FAQ mogą dotyczyć odizolowanych od siebie pytań i odpowiedzi na dany temat, bądź też mogą mieć bardziej określoną ustaloną strukturę typu pytanie/odpowiedź wewnątrz danej grupy tematycznej[90].

Użytkownik pragnący skorzystać z bogatej biblioteki zbiorów FAQ (zbiorów pytań i odpowiedzi) może mieć poważne trudności ze znalezieniem właściwych informacji. Inteligentni agenci FAQ indeksują zbiory typu FAQ i dostarczają użytkownikowi odpowiedniego interfejsu pozwalającego na formułowanie pytań w języku naturalnym. Systemy te wykorzystują w przeszukiwaniu ciąg znaków występujący w pytaniu. W przeciwieństwie do systemów indeksujących agenci FAQ uzyskują odpowiedzi wyłącznie na pytania, które pojawiają się w poindeksowanych zbiorach FAQ. Dzieje się tak z powodu częściowo tylko ustrukturalizowanych zbiorów typu FAQ oraz faktu, że zbiorów typu FAQ jest mniej niż

[88] Wg [ETW95].

[89] Patrz http://www.cs.washington.edu/research/metacrawler/.

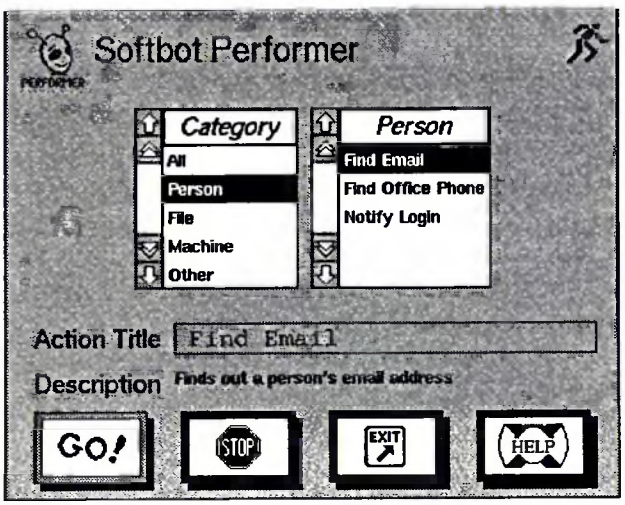
[90] Wg [ETW95].

dokumentów WWW. Przyjmuje się też, że inteligentni agenci FAQ są bardziej wiarygodni niż agenci indeksujący.

Systemy poszukujące wiedzę ekspertów – system Softbot

System Softbot z Uniwersytetu w Waszyngtonie (ang. *University of Washington*) jest ambitną próbą stworzenia systemu, którego zamierzeniem jest zrozumienie potrzeb użytkownika (tzn. odpowiednio zinterpretować jego zapytanie(a)) i kontekst dostarczanych informacji. System Softbot dokonuje analiz różnego rodzaju informacji dostępnych w sieci Internet (różnego rodzaju serwery, serwisy *NetFind, Federal Express's package tracking service* i inne). Ponieważ informacja, której dostarcza system Softbot jest ustrukturalizowana, dlatego system nie musi wykorzystywać języków naturalnych, czy języków wyszukiwawczych, aby „zrozumieć" dostarczone informacje. W rezultacie Softbot jest w stanie udzielić odpowiedzi na dość wąskie pytania ze stosunkowo dużym stopniem dokładności.

Kluczowa idea systemu Softbot streszcza się w jego nazwie (pochodzi od ang. *Software Robot*). System dostarcza szeregu usług programowych jak np. ftp, wydruk, poczta elektroniczna czy usług związanych z siecią Internet (jak np.: *Finger, NetFind*).



Rys. 2.24 System Softbot

Cel systemu Softbot

System jest prototypem systemu doradczego wysokiego poziomu. Przejmuje ogólnie pojęte cele użytkownika i w sposób dynamiczny syntetyzuje właściwą sekwencję poleceń wykonywanych w sieci Internet. Softbot wykonuje odpowiednie czynności zbierając informacje w celu podjęcia ostatecznych decyzji (czasami powtarzając niektóre polecenia). Wymienia się następujące rodzaje zadań, jakie wykonuje Softbot:

1/ powiadomienie użytkownika,
2/ wprowadzanie ograniczeń,
3/ lokalizacja i zarządzanie obiektami,

1. *Powiadomienie użytkownika* oznacza, że system jest w stanie wykonywać szereg czynności, jak np.: zarządzanie zbiorami na dysku, usługi typu BBS (ang. *Bulletin Board System*), FTP powiadamiając użytkownika np. sygnałem dźwiękowym.

2. *Wprowadzanie ograniczeń* oznacza, że Softbot jest w stanie działać w otoczeniu konkretnego systemu operacyjnego, np.: zapytując użytkownika o atrybuty zbiorów na dysku czy kompresję zbiorów.

3. *Lokalizacja i zarządzanie obiektami* oznacza, że system jest w stanie dokonywać konwersji dokumentów źródłowych i uzyskiwać dostęp do odległych baz danych. W momencie, gdy użytkownik uznał za stosowne komunikować się z określonymi ludźmi, obiektami, czy wybranymi zasobami informacji, system Softbot automatycznie generuje odpowiedni ciąg poleceń aż do momentu usunięcia wieloznaczności w żądaniu użytkownika. Np. żądanie wydruku danego zbioru powoduje, że system zacznie ustalać rozmiar kolejki wydruku (liczba zadań) i status (priorytet). Podobnie w momencie podania nazwy danego użytkownika, system rozpocznie poszukiwania jego adresu elektronicznego przez dostępne serwisy sieci Internet (*Whois, NetFind, staffdir, Finger* itp.).

Oczywiście powyższa lista nie jest wyczerpująca, ale ilustruje sposób, w jaki Softbot działa. System Softbot pozostawia użytkownikowi jedynie wybór celu działania natomiast samodzielnie podejmuje decyzje, jak wykonać dane działanie i gdzie można uzyskać odpowiedź na pytanie.

Określanie celów

Określanie celów jest szczególnie użyteczne wtedy, gdy użytkownikowi jest łatwiej określić cel, niż dokonać jego realizacji. Aby określenie celów było przydatne dla użytkownika, system Softbot musi spełniać trzy kryteria:

1/ posiadanie języka pozwalającego na formułowanie celów,
2/ *GUI* (ang. *Grafical User Interface*) – graficzny interfejs użytkownika,
3/ dialog systemu z użytkownikiem.

Pierwszy element oznacza możliwość określenia w sposób przyjazny dla użytkownika celów, dlatego proponuje się zastosowanie operatorów: alternatywy, koniunkcji, negacji.

Mimo dużych możliwości operatorów logicznych istnieje obawa, że ich stosowanie napotka na opory ze strony wielu użytkowników zwłaszcza w przypadku zbyt złożonych wyrażeń, dlatego system Softbot posiada wygodny graficzny interfejs użytkownika (punkt 2) do formułowania złożonych pytań. Na razie nie wprowadzono interfejsu wykorzystującego język naturalny.

Ostatnim elementem omawianego systemu Softbot jest możliwość odpowiedniego dialogu użytkownika z systemem (punkt 3). Moduł dialogu pozwala użytkownikowi wprowadzać dodatkowe ograniczenia wyszukiwawcze.

Architektura systemu Softbot

System składa się z czterech podstawowych modułów:

- *Moduł zadań* (ang. *task manager*)
- *Moduł planowania* (ang. *XII planner*)
- *Moduł zarządzania modelami* (ang. *model manager*)
- *Moduł internetowych domen* (ang. *Internet domain models*)

Moduł zadań kontroluje wszystkie istotne działania systemu Softbot począwszy od działań na poziomie planowania a skończywszy na działaniach na poziomie operacyjnym (np. łączenie się z serwerami Gopher).

Moduł planowania pozwala na planowanie dalszych działań przy pomocy niekompletnych danych. Moduł ten analizuje dostępne plany działania dopóki nie zostanie znaleziony odpowiedni plan pozwalający na osiągnięcie zamierzonego celu.

Moduł zarządzania modelami jest wyspecjalizowaną bazą danych, która przechowuje wszystko, co zostało zaobserwowane przez system. Istotnym aspektem modułu zarządzania zbiorami jest możliwość rozumowania wykorzystującego założenie o zamkniętości świata (ang. *Closed World Assumption*). Rozumowanie przyjmujące takie założenie daje możliwość wyciągania wniosków opartych na stwierdzeniu, że wiadomo wszystko o istnieniu wszystkich istotnych obiektów. Rozumowanie tego typu stanowi podstawę systemu kierowanego celami, jakim jest Softbot[91].

Moduł internetowych domen dostarcza systemowi Softbot szeregu informacji o samej sieci Internet, jej bazach danych, usługach itp. Moduł ten zawiera również heurystyki wyszukiwawcze oraz dane na temat ludzi (zawód, numer telefonu, adres e-mail) oraz wybranych komputerów (lokalizacja, charakterystyka systemu).

System Softbot posiada wiele, ale nie wszystkie cechy agentów omówionych wcześniej. Jest on wysoce autonomiczny; kierowany celami i potrafi dostosować się do zachodzących zmian.

---

[91] Oczywiście sieć Internet jest na tyle rozległa, że system Softbot nie może przyjąć, że zna zawartość każdej bazy danych na każdym komputerze bazowym.

3. MODEL SAMOUCZĄCEGO SIĘ MECHANIZMU DOSTĘPU DO SIECI

3.0. WSTĘP

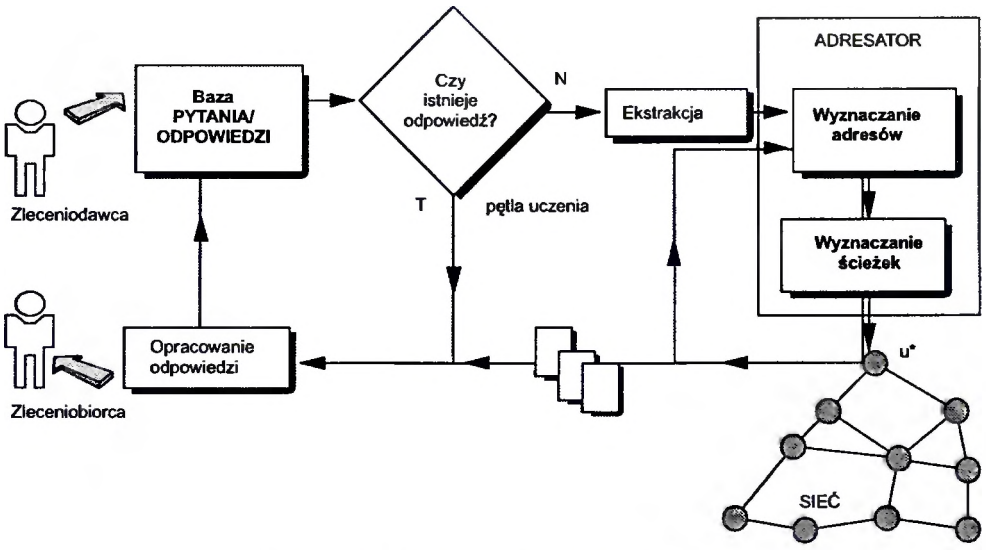
W rozdziale poprzednim przedstawione zostały zasady organizacji i działania złożonych sieci informacyjnych. Z punktu widzenia użytkownika istotnym aspektem w procesie korzystania z sieci jest zapewnienie dostępu do tej części jej zasobów, która może zawierać poszukiwane informacje. W przypadku dużych, rozległych i rozproszonych sieci kwestia ta nabiera szczególnego znaczenia z uwagi na efektywność procesu lokalizowania zasobów, mierzoną czasem potrzebnym na odnalezienie właściwych serwerów w sieci i dotarciem do zbiorów danych oraz stopniem relewantności wyników.

W rozdziale tym przedstawiony zostanie formalny model mechanizmu dostępu do sieci. Potrzeba zbudowania takiego modelu wynika z faktu, że kwestia dostępu do sieci, nawigacji w sieci i docierania do poszukiwanych zasobów jest bardzo złożona, gdyż składają się na nią problemy organizacyjne, techniczne i językowe. Model ma za zadanie wydobyć i przebadać ze złożonej rzeczywistości „sieciowej" te jej aspekty, które są istotne z punktu widzenia celu niniejszej pracy. Chodzi więc o uproszczenie obszaru badanego świata, ale z zachowaniem istotnych jego składników i cech. Pamiętać przy tym należy o przestrodze Einsteina: *rzeczy powinny być tak proste, jak to możliwe, ale nie prostsze.*

Zaproponowany model ma własność uczenia się na podstawie realizowanych przez siebie zadań. Naturalnie termin „uczenie się" ma w kontekście tej pracy ograniczone znaczenie. Tutaj uczenie się polega przede wszystkim na powiększaniu przechowywanej w modelu ilości informacji faktograficznych o zasobach i strukturze sieci. Wiedza o rozmieszczeniu zasobów sieci i ich zawartości powiększa się w miarę wykonywanych zadań.

3.1. ZAŁOŻENIA

Przed przystąpieniem do omówienia założeń modelu zostanie naszkicowany proces obsługi zleceń kierowanych do sieci. Schemat podany na rys. 3.1. zawiera główne elementy tego procesu. Poniżej zostaną one pokrótce przedyskutowane.



Rys. 3.1 Schemat procesu dostępu do sieci

Proces rozpoczyna użytkownik od sformułowania zlecenia do sieci. Najczęściej zlecenie jest pytaniem (zleceniem) kierowanym do zasobów informacyjnych sieci w intencji otrzymania poszukiwanych informacji. Niekiedy są to jednak zlecenia „per se", np. polecenie wydrukowania zawartości wskazanego katalogu. W omawianym tu modelu żądania kierowane do systemu zostaną ograniczone do zleceń, które są pytaniami.

Przed rozpoczęciem właściwej fazy opracowywania pytania i przesyłania go do sieci, zlecenie jest kierowane do bazy *PYTANIA/ODPOWIEDZI*, gdzie znajdują się pytania wcześniej skierowane do sieci wraz z odpowiedziami.

Jeśli dane pytanie można obsłużyć za pomocą bazy *PYTANIA/ODPOWIEDZI* i otrzymany wynik satysfakcjonuje zleceniodawcę, wówczas proces obsługi zlecenia można zakończyć bez faktycznego dostępu do sieci. W przeciwnym razie zlecenie zostanie opracowane przez mechanizmy zawarte w modelu i skierowane do sieci.

Pierwszym krokiem przy opracowaniu zlecenia jest ekstrakcja, która polega na wydobyciu z pytania wszystkich słów kluczowych, nazywanych także termami. Następnie słowa te kierowane są do tzw. adresatora, gdzie wyznaczane są adresy tych zasobów sieci, które mogą zawierać informacje relewantne do pytania. Na podstawie tych adresów realizowany jest dostęp do sieci, co polega na wyznaczeniu ścieżek dostępu do serwerów i zbiorów wskazanych przez adresator. Teraz można już „wydobyć" z sieci poszukiwane informacje i „sprowadzić je do modelu". Na rysunku 3.1. sieć przedstawiona jest symbolicznie za pomocą grafu, którego węzłami są serwery. Jeden z nich oznaczony przez **u***, ma szczególne znaczenie, jest to bowiem ten serwer, z którym związany jest rozważany model dostępu.

Informacje uzyskane z sieci wykorzystywane są w dwojaki sposób: po pierwsze – stanowią podstawę do opracowania odpowiedzi, która następnie przekazywana jest zleceniobiorcy i po drugie – są przekazywane do adresatora (na rysunku 3.1 jest to pętla uczenia) zwiększając jego wiedzę o sieci. Efekt ten nazywamy tutaj samouczeniem się mechanizmu dostępu do sieci.

Założenia modelu

Oto założenia, jakie przyjęto przy opracowywaniu modelu przedstawionego w następnych punktach tego rozdziału:

- *zlecenia,* czyli pytania, są wyrażeniami złożonymi z termów (które są słowami kluczowymi). Termy są połączone operatorami boolowskimi *I, LUB*. Należy zaznaczyć, że z modelu wyłączono operator *NIE*, głównie ze względu na zakładaną rozległość zasobów. Zostanie to wyjaśnione na przykładzie: wykonanie prostego zlecenia typu „*NIE* Geografia" spowodowałoby wskazanie jako właściwych wszystkich zasobów za wyjątkiem tych, które indeksowane są terminem „Geografia". W dużych sieciach i zasobach spowodowałoby to „potop informacyjny".
  Naturalnie niebezpieczeństwo „potopu informacyjnego" istnieje także w przypadku pytań wykorzystujących operatory *LUB, I*. Jest ono jednak mniejsze, zwłaszcza że nawiązuje ono lepiej niż operator *NIE* do intuicji przeciętnego użytkownika;
- konwencja indeksowania zbiorów informacyjnych sieci jest hierachiczna i składa się z następujących elementów rozdzielonych kropką:
  Adres_Serwera. Nazwa_Zbioru. Słowo_kluczowe,
  gdzie *Adres_Serwera* identyfikuje serwer (komputer bazowy), na którym przechowywany jest plik, którego nazwę podaje *Nazwa_Zbioru,* zaś *Słowo_Kluczowe* opisuje zawartość pliku;
- centralnym elementem modelu jest adresator, który swą budową przypomina prosty tezaurus: zawiera on termy wraz z określonymi na nich relacjami. W zbiorze tych relacji wyróżniono relacje: synonimii, „węższe", „szersze" i dla tych właśnie relacji przeprowadzono dalsze rozważania, w szczególności dotyczące oceny dokładności strategii dostępu do sieci. Należy podkreślić jednak, że nic nie stoi na przeszkodzie w dowolnym zwiększaniu liczby tych relacji[92];
- ocenę dokładności strategii dostępu oparto na arbitralnie wytypowanej funkcji (fragment paraboli), która dobrze spełnia przyjęte założenia (wymagania) co do tego typu funkcji. Jest przedmiotem eksperymentu sprawdzenie, czy rzeczywiście funkcja ta daje dobre rezultaty i w przypadku odpowiedzi negatywnej na to pytanie zaprojektowanie innej funkcji;
- dokładność strategii dostępu porównywana jest z wartością progową $0<\varepsilon\leq 1$, arbitralnie ustaloną przez administratora systemu. Trzeba podkreślić, że wartość ε można uzyskać na drodze eksperymentu z systemem. W pierwszej fazie cyklu

[92] Np. przez wprowadzenie terminów kojarzeniowych.

życia systemu za ε należy przyjąć stosunkowo niewielką wielkość. W miarę uczenia się systemu wartość ε należy zwiększać;

- przyjęto przyrostowy (monotoniczny) model uczenia się systemu, co oznacza, że „wiedza" systemu o topologii sieci i jej zasobach wzrasta w miarę jego eksploatacji i nie może się zmniejszać. Wiedza systemu mierzona jest liczbą termów zawartych w *adresatorze,* liczbą adresów serwerów i zbiorów związanych z tymi termami oraz liczbą relacji określonych na termach w adresatorze.

3.2. SIEĆ

Definicja 3.1 (sieci)

Parę uporządkowaną **S=(U, χ)** nazywa się **siecią S,** gdzie:

U – jest skończonym zbiorem, którego elementy nazywane są serwerami; zakładamy, że
$\mathbf{U} \neq \emptyset$

χ – jest dwuargumentową relacją określoną na zbiorze **U**, tzn. $\chi \subseteq \mathbf{U} \times \mathbf{U}$, nazywaną konfiguracją sieci; zakładamy, że **χ** jest relacją spełniającą warunki:

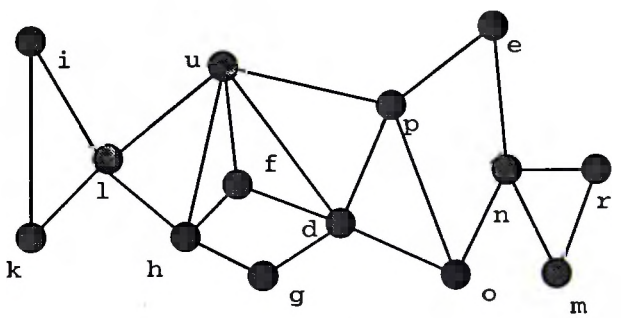
$\forall\, \mathbf{u}', \mathbf{u}'' \in \mathbf{U}\ \mathbf{u}'\ \chi\ \mathbf{u}'' \leftrightarrow \mathbf{u}''\ \chi\ \mathbf{u}'$ (symetria)

$\forall\, \mathbf{u} \in \mathbf{U}\ \mathbf{u}\ \chi\ \mathbf{u}$ (zwrotność)

■

W sieci wyróżnia się jeden serwer, oznaczany przez **u***. Jest to serwer, z którym bezpośrednio komunikuje się wskazany użytkownik sieci. Serwer ten stanowi „wejście" do sieci dla tego użytkownika. Z matematycznego punktu widzenia sieć jest grafem (nieskierowanym)[93]. Serwery są węzłami tego grafu zaś relacja **χ** określa jego krawędzie. Przez $\mathbf{K}=2^{\mathbf{U}\times\mathbf{U}}$ oznaczany jest zbiór wszystkich konfiguracji sieci.

Przykład



Rys. 3.2 Przykład sieci

[93] Wg [BOR77]

Przykładem sieci zgodnie z def. 3.1. jest następujący zbiór komputerów (rys. 3.2.):

U={i, k, l, u, h, f, d, g, p, n, o, e, r, m} wraz z określoną na nich relacją χ.

Należy odnotować, że dla zwykłego użytkownika sieć w znaczeniu podanym w definicji 3.1. jest przezroczysta (czyli komputery pośredniczące w dostępie do informacji są niewidoczne).

▲

Definicja 3.2 (dołączanie serwera)

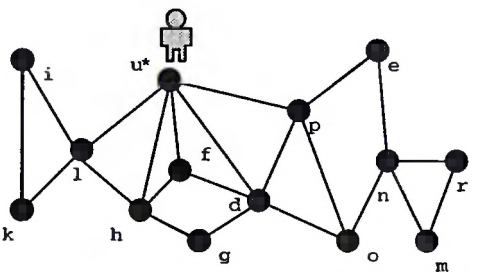
Dla sieci **S=(U, χ)**, konfigurację χ nazywa się **konfiguracją z dołączeniem serwera u*:**

u* $\in$ **U** do sieci **S**, jeśli istnieje **u** $\in$ **U** i **u*** $\neq$ **u** takie, że **u*** χ **u**

■

Przykład

Rysunek 3.3 przedstawia schematycznie dołączanie serwera **u***. W przypadku sieci komputerowych dołączanie serwera oznacza dostęp użytkownika do dowolnego komputera pracującego w sieci[94].



Rys. 3.3 Dołączanie serwera

Definicja 3.3 (funkcji adresowej)

Niech **A** będzie skończonym zbiorem adresów, zaś zbiór **U** zbiorem serwerów sieci **S**. Funkcję α określoną jako:

$$\alpha : U \xrightarrow[1\text{-}1]{w} A$$

Powyższą funkcję nazywamy **funkcją adresową sieci S.** Wartość tej funkcji **α(u)**, dla **u** $\in$ **U**, nazywamy adresem serwera **u**.

■

[94] Por. serwer u* na rys. 3.1 i 3.3.

Należy zauważyć, że ze względu na jednoznaczność funkcji α można podać definicję $S=(U,\chi)$ również jako $S=(\alpha^{-1}(A), \chi)$. Obie definicje są równoważne. Oznacza to w praktyce, że adres serwera jednoznacznie określa (identyfikuje) serwer w sieci.

Przez $\mathbf{Z}^{n}$ oznaczamy skończony zbiór zasobów, na przykład baz danych. Zasoby ze zbioru $\mathbf{Z}^{n}$ ulokowane są w serwerze $u \in U$.

Przez $z=\bigcup_{i-1}^{n} z^{n_i}$

oznaczamy wszystkie zasoby ulokowane w sieci **S** złożonej z serwerów $\{u_1,...,u_n\}$. Zasobom przyporządkowujemy nazwy ze skończonego zbioru nazw **R**.

Przykład

Wszystkie komputery pracujące w sieci Internet mają niepowtarzalne adresy. Na przykład 148.81.213.2 jest niepowtarzalnym adresem jednego z komputerów bazowych w sieci Internet. (często zastępowany adresem alfanumerycznym np.: novell.ibin.uw.edu.pl.) Przyporządkowanie unikalnych adresów komputerom (nie tylko serwerom) pracującym w sieci Internet jest warunkiem poprawnego funkcjonowania sieci wykorzystującej protokół TCP/IP.

Definicja 3.4 (funkcji nazw zasobów)

Funkcję β określamy jako:

$$\beta: Z \xrightarrow[n-1]{w} R$$

taką, że:

$\forall\, u \in U\ \forall z', z'' \in Z^u\ z' \neq z'' \leftrightarrow \beta(z') \neq \beta(z'')$

nazywamy **funkcją nazwy zasobów.** Wartość tej funkcji $\beta(z')$, dla $z \in Z^u$, nazywamy nazwą zasobu **z**.

Zauważmy, że przy tak zdefiniowanej funkcji różne zasoby mogą mieć te same nazwy, pod warunkiem, że są rozmieszczone w różnych serwerach. W dalszym ciągu będzie stosowana konwencja „kropkowa" (obok konwencji URL) na określenie nazwy zasobu ulokowanego na serwerze **u**:

$$\alpha(u) \,.\, \beta(z)$$

Przykład

Według konwencji kropkowej:

148.81.213.4 . mmwwwpc/mmwwwpc.html . Polska

Coraz powszechniej jednak jest stosowany tzw. URL (ang. *Uniform Resource Locator*) oznaczający oprócz adresu danego serwera również położenie zbioru (katalog) oraz nazwę zbioru:

http://adres.serwera.w_sieci/nazwa_katalogu/nazwa_zbioru.html
np.: htttp://www.univ.trieste.it/mmwwwpc/mmwwwpc.html.
gopher://plearn.edu.pl/bibn/info.pl//uw/inwy/hist/bibn/info.txt
ftp://148.81.213.4//sss4/aplik.zip

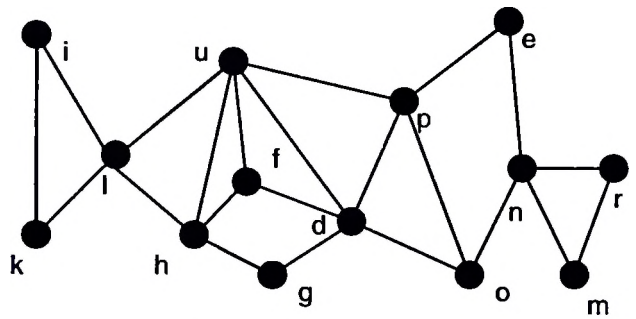
Definicja 3.5 (ścieżki)

Ścieżką łączącą serwery u_i , u_k w konfiguracji oznaczoną przez s_{i-k} nazywamy wektorem

$$s_{i-k}=<u_i,...,u_k> \text{ taki, że } u_j \chi u_{j+1}, j = i ,..., k-1, k$$

Przykład

Niech przykładem sieci S będzie rysunek 3.4 przedstawiający ścieżki dostępu łączące poszczególne komputery.



Rys. 3.4 Ścieżka łącząca serwery

Ścieżkami łączącymi serwery **i, k** mogą być następujące wektory:

$$s_{i-k}=< i, k >,$$
$$s_{i-k}=<, i, l, k >$$

Ścieżkami łączącą serwery **u, g** mogą być następujące wektory:

$$s_{u-g}=< u, h, g >,$$
$$s_{u-g}=< u, f, d, g >,$$
$$s_{u-g}=< u, d, g >,$$
$$s_{u-g}=< u, l, h, g > (...)$$

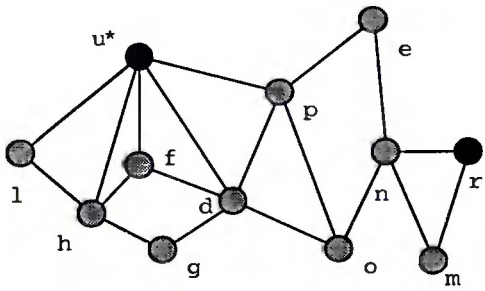
Ścieżkami łączącą serwery **r, g** mogą być następujące wektory:

$$s_{r-g}=< r, m, n, o, d, g >,$$
$$s_{r-g}=< r, n, e, p, d, g > (...)$$

Definicja 3.6 (ścieżki dostępu)

Ścieżkę łączącą serwery u^*, u_k w konfiguracji χ nazywamy **ścieżką dostępu** i oznaczamy przez $s_{i-k}=<u^*,...,u_k>$, lub przez $s_{*\_u_k}$.

Przykład



Rys. 3.5 Ścieżka dostępu

Według rysunku 3.5 ścieżkami dostępu łączącym serwer dostępu (por. def. 3.2.) **u*** i serwer **r** mogą być następujące wektory :

s_{u^*-r}=< **u*, p, e, n, r** >,
s_{u^*-r}=< **u*, p, o, n, r** >,
s_{u^*-r}=< **u*, p, o, n, m, r** >,
s_{u^*-r}=< **u*, f, d, o, n, m, r** >,
s_{u^*-r}=< **u*, h, g, d, o, n, r** >,
s_{u^*-r}=< **u*, h, g, d, o, n, m, r** > (...)

▲

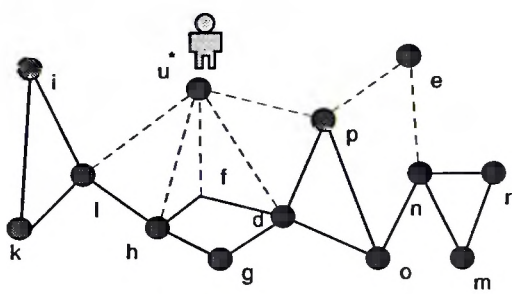
Definicja 3.7 (dostępności sieci)

Sieć S jest dostępna z u* jeśli istnieje takie **u** ∈ **U**, że istnieje ścieżka s_{*-u} , dla **u* ≠u.**

■

Przykład

Niech poniższy rysunek będzie negatywną ilustracją definicji 3.7. Linie przerywane oznaczają brak połączenia z serwerami.



Rys. 3.6 Dostępność sieci

Na przykładzie sieci z rys. 3.6 dostępność sieci z serwera **u*** nie jest możliwa, mimo że serwer ten należy do sieci. Jak widać sieć pokazana na rysunku 3.6. jest również niespójna. Stąd można zaryzykować twierdzenie, że sieć spójna będzie zawsze dostępna (o ile serwer dostępu należy do sieci), gdyż zawsze istnieje ścieżka dostępu.

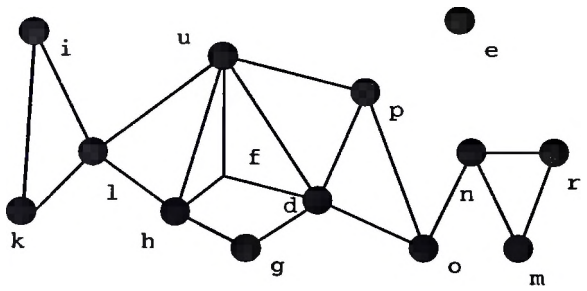
Definicja 3.8 (spójności sieci)

Sieć $S=(U, \chi)$ jest spójną wtedy i tylko wtedy, gdy spełniony jest warunek:
$\forall u \in U \; \exists u \in U \; u \neq u' \rightarrow u \chi u'$

■

Definicja ta mówi, że w sieci spójnej nie ma odizolowanych serwerów. Obecnie zostaną podane dwa ważne twierdzenia dotyczące spójności sieci.

Przykład



Rys. 3.7 Sieć niespójna

Sieć przedstawiona na rys. 3.7. nie jest spójna ze względu na serwer **e.**

Twierdzenie 3.1

Jeśli dla dowolnych węzłów $u_i, u_j \in U$ sieci S istnieje ścieżka s_{i-j}, to sieć S jest spójna. Prawdziwe jest również twierdzenie odwrotne.

Twierdzenie 3.2

Sieć S jest spójna wtedy i tylko wtedy, gdy dla każdego $u \in U$ sieć S jest dostępna z u.

Definicja 3.9 (podsieć)

Sieć $S'=(U', \chi')$ nazywamy podsiecią sieci $S=(U, \chi)$ wtedy i tylko wtedy, gdy $U' \subseteq U$ oraz $\chi' \subseteq \chi$, co zapisujemy jako $S' \subseteq S$.

■

Twierdzenie 3.3

Jeśli sieć S jest spójna, to każda podsieć $S' \subseteq S$ jest także spójna.

Obecnie podane zostanie kilka własności dotyczących operacji na sieciach. Własności te zostaną poprzedzone definicją tych operacji. Są to operacje: $\cup, \cap, \neg$ (sumy, iloczynu, dopełnienia).

Definicja 3.10 (sumy, iloczynu i dopełnienia na sieciach)

Niech S', S'' i S będą sieciami.

(1) $S'' \cup S'' = (U' \cup U'', \chi' \cup \chi'')$

(2) $S'' \cap S'' = (U' \cap U'', \chi' \cap \chi'')$

(3) Niech $S' \subseteq S$

$\neg S' = (U \setminus U', \chi \setminus \chi')$

■

Twierdzenia dot. sumy, iloczynu i dopełnienia na sieciach

(1) $S', S'' \subseteq S \rightarrow S' \cup S'' \subseteq S$

(2) $S', S'' \subseteq S \rightarrow S' \cap S'' \subseteq S$

(3) Jeśli S', S'' są sieciami spójnymi, to $S' \cap S''$ jest także siecią spójną.
Uwaga: własność (3) nie zachodzi dla operatora $\cup$.

(4) Jeśli $S' \subseteq S$ i S jest siecią spójną, to $\neg S'$ jest podsiecią spójną.

3.3. ZLECENIA

Definicja 3.11 (termu)

Termem nazywamy skończony ciąg znaków alfanumerycznych, włącznie ze znakiem spacji ("puste")

■

Przykład

Przykładami termów są:

Geografia,

nauki polityczne,

Ruch na Rzecz Wolności Roślin.

▲

Niech T będzie skończonym zbiorem termów. Obecnie zostanie zdefiniowane pojęcie języka nad zbiorem termów.

Definicja 3.12 (języka)

Trójka uporządkowana **L=(T,O,R),** gdzie:

T – jest zbiorem termów

O – jest skończonym zbiorem operatorów

R – jest skończonym zbiorem reguł budowania wyrażeń.

Uwaga: dalsze rozważania ograniczone będą do operatorów $\wedge$, $\vee$

Zbiory te złożony są z pięciu reguł generowania wyrażeń, a mianowicie:

(I) jeśli $\mathbf{t} \in \mathbf{T}$, to **t** jest wyrażeniem,

(II) jeśli $\mathbf{w}'$, $\mathbf{w}''$, to napis $\mathbf{w}' \vee \mathbf{w}''$ jest wyrażeniem,

(III) jeśli $\mathbf{w}'$, $\mathbf{w}''$, to napis $\mathbf{w}' \wedge \mathbf{w}''$ jest wyrażeniem,

(IV) jeśli **w,** to napis **(w)** jest także wyrażeniem,

(V) tylko napisy utworzone zgodnie z regułami **(I)-(IV)** są wyrażeniami.

■

Przykład

Niech **T** = {geografia, Polska, ochrona środowiska}
Wyrażeniami są m.in. następujące napisy:

Polska $\wedge$ ochrona środowiska

geografia $\wedge$ Polska $\wedge$ ochrona środowiska

(geografia $\vee$ ochrona środowiska) $\wedge$ Polska

▲

Jeśli **w** jest wyrażeniem utworzonym zgodnie z regułami podanymi w definicji 3.12, to **w** nazywa się wyrażeniem języka **L,** co jest zapisywane jako $\mathbf{L} \vdash \mathbf{w}$. Można również powiedzieć, że **w** należy do **L.**

Definicja 3.13 (subjęzyk)

Niech **W** będzie zbiorem wyrażeń. Zbiór **W** nazywany jest subjęzykiem języka **L,** jeśli spełniony jest warunek :

$$\forall \mathbf{w} \in \mathbf{W}\ \mathbf{L} \vdash \mathbf{w}$$

Należy zauważyć, że zbiór termów występujący w określeniu języka **L** jest z definicji subjęzykiem języka **L.**

W rozumieniu definicji 3.12 język jest wyłącznie konstrukcją syntaktyczną. Naturalnie w praktyce użytkownicy języka przyporządkowują znaczenia wyrażeniom języka. tzn. przyporządkowują znaczenia termom i operatorom. Zagadnienie semantyki jest bardzo złożone i nie będzie omawiane. Zostaną jednak poczynione pewne ustalenia w celu lepszego przedstawienia i uporządkowania rozważanych dalej kwestii.

Niech **M** oznacza skończony zbiór znaczeń termów, zaś **T** będzie zbiorem termów języka **L.**

Definicja 3.14 (znaczenia)

Funkcja ρ określona jako:

$$\rho : T \xrightarrow{w} M$$

będzie nazywana znaczeniem języka **T** w **M.**

■

Dla $t \in T$, zapis $\rho U(t)$ rozumiany jest jako znaczenie termu **t.** Powyższa definicja dopuszcza istnienie synonimów (różne termy o tym samym znaczeniu), ale nie dopuszcza istnienia różnych znaczeń tego samego termu. Teraz zostaną zdefiniowane formalnie synonimy.

Definicja 3.15 (synonimu)

Termy $t', t'' \in T, t' \neq t''$ nazywa się synonimami w języku **T,** jeśli

$$T \vdash t', T \vdash t'' \rightarrow \rho(t') = \rho(t'')$$

■

Zdefiniowanie pojęcia *„term* **t'** *o znaczeniu szerszym niż znaczenie termu* **t''"** oraz pojęcia *„term* **t'** *o znaczeniu węższym niż znaczenie termu* **t''"** jest w ramach stosowanej tu formalizacji trudne. Dlatego przyjęte zostanie jako intuicyjnie zrozumiałe. Na przykład term *CIECZ* ma znaczenie szersze niż term *WODA;* term *SILNIK DIESLA* ma znaczenie węższe niż term *SILNIK.*

Zostanie przyjęta konwencja, że jeśli term **t'** jest znaczeniowo szerszy od **t''**, to będzie to zapisywane jako:

$$\rho(t') \succ \rho(t'')$$

Podobnie, jeśli **t'** jest znaczeniowo węższe od **t''**, to będzie to zapisywane następująco:

$$\rho(t') \prec \rho(t'')$$

Na zbiorze termów **T** zostaną zdefiniowane następujące relacje dwuargumentowe („szersze", „węższe", „równe").

Definicja 3.16 (relacji „szersze")

I. $B \subseteq T \times T$ taka, że $\forall t', t'' \in T$
$t' B t'' \Leftrightarrow \rho(t') \succ \rho(t'')$

Definicja 3.17 (relacji „węższe")

II. $N \subseteq T \times T$ taka, że $\forall t', t'' \in T$
$t' N t'' \Leftrightarrow \rho(t') \prec \rho(t'')$.

Definicja 3.18 (relacji „równe")

III. $S \subseteq T \times T$ taka, że $\forall t', t'' \in T$
$t' S t'' \Leftrightarrow \rho(t') = \rho(t'')$.

Relacje te mają następujące własności:

- relacje **B, N** są przechodnie
- relacja **S** jest relacją równoważności
- $B=N^{-1}$, $N=B^{-1}$

Własności funkcji znaczenia wyrażeń

W uzupełnieniu rozważań o semantyce termów należy zwrócić uwagę na fakt, że podobnie można określić znaczenie wyrażeń. Pomijając kroki przygotowawcze i przyjmując od razu, że jest funkcją nadającą wyrażeniom znaczenia można odnotować, że funkcja ta ma następujące własności:

$$(I)\ \lambda(w'' \wedge w') = \lambda(w') \wedge \lambda(w'')$$

$$(II)\ \lambda(w'' \vee w') = \lambda(w') \vee \lambda(w'')$$

dla **w''**, **w'** należących do pewnego języka **W.**
Niech **w** będzie wyrażeniem języka $L = (T, O, R)$, tzn. $L \vdash w$.

Definicja 3.19 (ekstrakt)

Ekstraktem wyrażenia **w,** oznaczanym przez **EX(w),** nazywamy zbiór wszystkich termów $t \in T$, które występują w wyrażeniu **w.**

Przykład

EX(w) (Zanieczyszczenie środowiska ∧ Polska) = {Zanieczyszczenie środowiska, Polska}

▲

Definicja 3.20 (zlecenie do sieci)

Zleceniem do sieci **S** nazywamy dowolne wyrażenie **w.**

■

3.4. OBSŁUGA ZLECEŃ

Obecnie zostanie naszkicowany proces (obsługi) zlecenia, a następnie zostaną bliżej omówione jego fazy. Przyjęto założenie, że zlecenie jest wyrażeniem przygotowanym przez użytkownika sieci.

1. Wyznaczenie ekstraktu zlecenia, czyli wyznaczanie **EX(w).**
2. Ewaluacja ekstraktu.
3. Poszukiwanie tych serwerów i ich zasobów, gdzie znajdują się dane relewantne do zlecenia. Sformułowanie to oznacza wyznaczenie podsieci złożonej z serwerów zawierających poszukiwane informacje.
4. Realizacja dostępu do serwerów, gdzie ulokowane są dane.
5. Zwiększanie wiedzy o sieci, tzn. o jej zasobach i ich rozmieszczeniu w wyniku realizacji dostępu. Faza ta będzie nazywana *uczeniem się modelu.*

Ekstrakcja jest operacją prostą określoną w definicji 3.19. Obecnie zostanie omówiony punkt 2, przedstawiony zostanie najważniejszy mechanizm modelu, tzw. adresator oraz pokazane zostanie, jak na jego przykładzie ewaluowane są zlecenia oraz jak realizowany jest dostęp do sieci.

Następnie przedstawione zostanie ważne twierdzenie dotyczące obsługi zleceń. Rozważania zostaną zakończone pokazaniem, jak przebiega proces uczenia się adresatora.

Definicja 3.21 (sygnatury)

Zostało przyjęte założenie, że każdy zasób (np. zbiór danych) ulokowany w serwerze **u** jest indeksowany zbiorem termów. Suma logiczna zbioru termów dla wszystkich zasobów tego serwera nazywana jest jego sygnaturą i oznaczana przez **SYGu.**

■

Przykład

Niech przykładem sygnatury dla serwera **u** będzie tabela 3.1.

Tabela 3.1 Przykład sygnatury dla serwera u

Term indeksujący	Nazwa Zbioru
Polska	/htdocs/tekst.htm
Polonia	/htdocs/tekst1.htm
Niemcy	/htdocs/tekst1.htm
Europa Wschodnia	/htdocs/tekst1.htm
Wspólnota Europejska	/htdocs/tekst2.htm
Belgia	/htdocs/tekst1.htm
Francja	/htdocs/tekst1.htm
Niemcy	/htdocs/tekst1.htm
Polska	/htdocs/tekst1.htm
Poland	/htdocs/tekst1.htm

▲

Definicja 3.22 (sieć relewantna dla termu t)

Podsiecią relewantną dla termu **t** nazywa się sieć **S=(U, χ)** taką, że dla każdego **u** $\in$ **U, t** $\in$ **SYG$_u$.**

■

Przykład

Niech podsiecią S relewantną dla termu t= *geografia* będzie zbiór serwerów:

{p, r, s}

Należy zaznaczyć, że poniższe tabele nie reprezentują sygnatur serwerów **p, r, s,** a jedynie te zbiory, które są indeksowane termem *geografia.*

Tabela 3.2. Zbiór dokumentów dla serwera p indeksowanych termem t

Term indeksujący	Nazwa Zbioru
geografia	/htdocs/tekst_g.htm
geografia	/htdocs/tekstg.htm

Tabela 3.3. Zbiór dokumentów dla serwera r indeksowanych termem t

Term indeksujący	Nazwa Zbioru
geografia	/htdocs/tekst_g.htm
geografia	/tekst_g.txt
geografia	/htdocs/tekst_g.htm

Tabela 3.4. Zbiór dokumentów dla serwera s indeksowanych termem t

Term indeksujący	Nazwa Zbioru
geografia	/htdocs/tekst_g.htm
geografia	/tekst_g.txt

▲

Definicja 3.23 (adresator)

Adresatorem nazywamy czwórkę uporządkowaną

ADR=(T, A, {B,N,S}, τ), gdzie:

T – jest zbiorem termów; $T \neq \emptyset$;

A – jest zbiorem adresów serwerów sieci (ewentualnie rozszerzonych nazwami zasobów, konwencja kropkowa", def. 3.4)

{B,N,S} – są relacjami: „szersze", „węższe", „synonimii" określonymi wyżej w zbiorze **T**

τ – jest relacją adresującą $\tau \subseteq T \times 2^A$ mającą własność:

$\forall\ t \in T\ \forall A' \subseteq A\ \forall\ a \in A'\ (u\ \tau\ A' \rightarrow t \in SYG_{\alpha^{-1}(a)})$, gdzie α jest funkcją adresową (z def. 3.3.).

■

Adresator jest zatem zbiorem termów, na którym określono relację i ponadto każdemu termowi **t** przyporządkowano zbiór adresów tych serwerów sieci, których sygnatury zawierają ten term.

Każdy element adresatora jest więc parą: term-zbiór adresów serwerów, gdzie znajdują się zasoby indeksowane tym termem.

Przykład

Tak więc adresatorem może być następująca tablica:

Tabela 3.5

Term	Adres serwera	Nazwa zbioru i katalogu
Szwecja	148.81.213.81	/pub/geografia/pltekst.htm
Dania	148.81.213.16	/pub/info/pltekst.txt
Holandia	148.81.213.21	/pub/geografia/pltekst.htm
Francja	148.81.213.4	/pub/kultura/doc.htm
S(Polska)=Poland	148.81.213.6	/pub/info/pltekst.txt
S(Stany Zjednoczone)=USA	148.81.213.5	/pub/geografia/pltekst.htm
S(Niemcy)=RFN	148.81.213.7	/pub/info/pltekst.txt

B(Kanada)=Ameryka Północna	148.81.213.16	/pub/info/pltekst.txt
B(Korea)=Azja	148.81.213.21	/pub/geografia/pltekst.htm
B(Polska)=Europa Wschodnia	148.81.213.10	/pub/kult/pltekst.htm
N(Stany Zjednoczone)=Kalifornia	148.81.213.8	/pub/geografia/pltekst.htm
N(Polska)=Mazowsze	148.81.213.52	/pub/geografia/pltekst.htm

Objaśnienia:

S(Polska) – synonim wyrażenia Polska

B(Polska) – term szerszy w stosunku do termu Polska

N(Polska) – term węższy w stosunku do termu Polska

Ewaluacja termów i wyrażeń za pomocą adresatora

Zostanie teraz przedstawiona kwestia ewaluacji termów i wyrażeń za pomocą adresatora. Przyjęte jest założenie, że adresy stowarzyszone z termami adresatora dotyczą zasobów tzn. zawierają adres serwera i nazwę zasobu.

Jeśli term adresatora jest stowarzyszony z pustym zbiorem adresów, co odpowiada sytuacji, że nie znane są (lub nie istnieją) serwery (podsieć) opisane tym termem, to uzasadnione wydaje się zidentyfikowanie tych podsieci, które są relewantne do termów związanych z termem **t** relacjami: **B, N, S.**

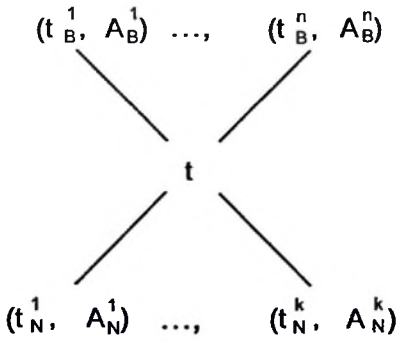
Należy zauważyć, że jeśli **t** jest związane jakimkolwiek termem **t'** przez relację synonimii oraz **t'** ma niepusty zbiór adresów, wówczas termy związane z **t** relacjami **B, N** są niepotrzebne.

Jeśli jednak term **t** nie ma synonimów lub ich zbiory adresów są również puste, to wykorzystywane są termy związane z **t** relacjami **B, N** wraz ze skojarzonymi z nimi zbiorami adresów w celu określenia podsieci „quasi relewantnej". Pojawia się jednak pytanie o dokładność takiego postępowania. Poniżej przedstawiona zostanie metoda obliczania dokładności.

Metoda obliczania dokładności (współczynnik aproksymacji)

Niech **||A||** oznacza moc zbioru **A,** tzn. liczbę jego wszystkich elementów. Zakłada się, że z termem **t** i wszystkimi jego synonimami związane są puste zbiory adresów. Natomiast istnieją termy szersze i węższe z niepustymi zbiorami adresów. Graficznie można to przedstawić za pomocą rysunku (rys. 3.8.)

W rys. 3.8 przyjęto następujące ustalenia: $\rho(t_B^j) \succ$ oraz $\rho(t_N^j) \prec$ **t, t =1,...,n; j = 1,...,k. t = 1,...,n; j = 1,...,k.**



Rys. 3.8 Zbiór wyrażeń węższych i szerszych w stosunku do wyrażenia t.

Do oceny dokładności przybliżenia termu posłużymy się współczynnikiem Θ nazywanym *współczynnikiem aproksymacji termu.* Przy projektowaniu współczynnika aproksymacji Θ przyjęto następujące założenia.

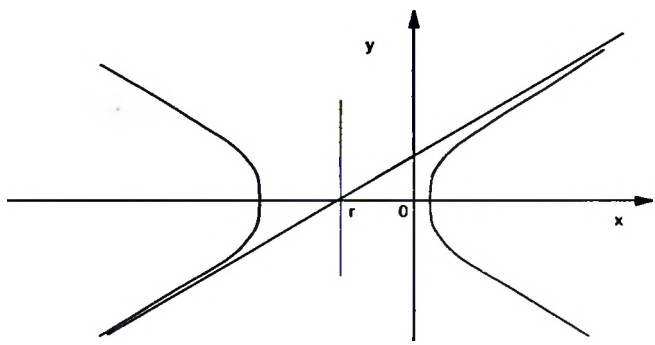
1) $0 \leq \Theta \leq 1$

2) Niech ℓ oznacza liczbę adresów zasobów, gdzie występują terminy związane relacjami **B,N** z termem **t.**

$$\lim_{\ell \to \infty} \Theta_t = 1$$

3) $\Theta_t \mid \ell = 0 = \varnothing$

Wydaje się, że dobrym przykładem matematycznym dla Θ jest wykres hiperboli $(x-r)^2-y^2=r^2$ odniesiony do jej asymptoty **y=x-r.**



Rys. 3.9 Wykres hiperboli jako przykład matematyczny współczynnika aproksymacji termu Θ_t.

A zatem dla termu **t** przy $\ell = \| A^1_B \cup ... \cup A^n_B \cup A^1_N \cup ... \cup A^k_N \|$ proponuje się następującą formułę na obliczanie współczynnika Θ_t.

$$\Theta_t = \frac{\sqrt{(\ell-1)^2-1}}{\ell-1}$$

W przypadku ekstraktu zlecenia **w**, tzn. **EX(w)={t1,...,tn}** współczynnik aproksymacji zlecenia można zdefiniować jako:

$$\Theta_w = \frac{\sum_{i=1}^{n} \Theta_i}{n},$$

Naturalnie możliwe są inne definicje podanych współczynników. Wszelkie propozycje w tym względzie powinny zostać zweryfikowane eksperymentalnie. Po wyznaczeniu wartości współczynnika aproksymacji należy podjąć decyzję, czy warto je kierować do sieci. W tej sprawie proponuje się arbitralne ustalenie wartości progowej $0<\varepsilon\leq 1$.

Jeśli $\Theta_w \geq \varepsilon$ wówczas zlecenie zostanie skierowane do sieci, w przeciwnym razie zostanie ono wstrzymane. Ustalenie wartości progowej ε można dokonać wyłącznie na drodze eksperymentów z siecią. Warto tu wszak odnotować, że w pierwszej fazie eksploatacji proponowanego tu modelu ε powinno być raczej niewielkie. W miarę uczenia się adresatora (patrz dalsza część wywodu) wartość ε należy zwiększać.

Niech **w** będzie zleceniem. Rozważony zostanie ekstrakt zlecenia **EX(w)={t1,...,tn}.** Niech **ADR=(T, A, {B,N,S}, τ)** będzie adresatorem. Obecnie wyznaczony zostanie zbiór wszystkich adresów stowarzyszonych z tymi termami ekstraktu, które występują w adresatorze. Przyjęte jest oznaczanie A_t^{τ} na zbiór adresów związanych z termem $t \in T$ relacją τ, tzn. $A_t^{\tau} = A$ wtedy i tylko wtedy, gdy $t \tau A$.

Definicja 3.24 (adresy stowarzyszone z termem t)

Zbiorem adresów stowarzyszonych z termem t należących do ekstraktu **EX** i występującym w adresatorze **ADR** (tzn. $t \in EX$ i $t \in T$) nazywany będzie zbiór określony następująco:

$$A_{t\in EX} = \begin{cases} A_t^{\tau}, \text{ gdy } A_t^{\tau} \neq \emptyset \\ \bigcup_{ti} A_{t_i}^{\tau}, \text{ gdy } t_i \in T, A_t^{\tau} = \emptyset \\ \quad \forall t_i \in T\, t_i St, \ \exists t_i A_{t_i}^{\tau} \neq \emptyset \\ \bigcup_{ti} A_{t_i}^{\tau}, \text{ gdy } t_i \in T, A_t^{\tau} = \emptyset \\ \quad \forall t_i \in T\, t_i Bt. \text{ lub } t_i Nt. \\ \exists t_i A_{t_i}^{\tau} \neq \emptyset \\ \emptyset \text{ w innych przypadkach} \end{cases}$$

■

Przykład

1. W pierwszym przypadku tzn., gdy A_t^{τ} jeśli $A_t^{\tau} \neq \emptyset$ przykładem zbioru adresów stowarzyszonych z termem **t** mogą być tabele 3.6. i 3.7.

Tabela 3.6 A_t^τ, jeśli $A_t^\tau \neq \emptyset$ dla t = **Francja**

Term	Adres serwera	Nazwa zbioru i katalogu
Francja	148.81.213.5	/pub/kultura/doc1.htm
Francja	148.81.213.4	/pub/kultura/doc.htm
Francja	148.81.213.4	/pub/kultura/doc.htm

Tabela 3.7 A_t^τ, jeśli $A_t^\tau \neq \emptyset$ dla t = **Holandia**

Term	Adres serwera	Nazwa zbioru i katalogu
Holandia	148.81.213.231	/pub/geografia/tekst.htm
Holandia	148.81.213.241	/pub/geografia/tekst.htm

2. W drugim przypadku tzn., gdy $A_t^\tau = \emptyset$ i istnieje t_i S t przykładem zbioru adresów stowarzyszonych z termem t może być tabela 3.8.

Tabela 3.8 $UA_{t_i}^\tau$ jeśli $A_t^\tau = \emptyset$ oraz t_iS t

Term	Adres serwera	Nazwa zbioru i katalogu
S(Polska)=Poland	148.81.213.6	/pub/info/pltekst.txt
S(Polska)=Polen	148.81.213.5	/pub/geografia/pltekst.htm
S(Polska)=Polen	148.81.213.7	/pub/info/pltekst.txt

3. W trzecim przypadku tzn., gdy $A_t^\tau = \emptyset$ i istnieje t_i B t przykładem zbioru adresów stowarzyszonych z termem t mogą być tabele 3.9. i 3.10.

Tabela 3.9 $UA_{t_i}^\tau$ jeśli $A_t^\tau = \emptyset$ oraz t_i B t

Term	Adres serwera	Nazwa zbioru i katalogu
B(Niemcy)=Wspólnota Europejska	148.81.213.6	/pub/info/pltekst.txt
B(Niemcy)=Europa Zachodnia	148.81.213.5	/pub/geografia/pltekst.htm
B(Niemcy)=Europa	148.81.213.7	/pub/info/pltekst.txt

Tabela 3.10 $UA_{t_i}^\tau$ jeśli $A_t^\tau = \emptyset$ oraz t_i N t

Term	Adres serwera	Nazwa zbioru i katalogu
N(Niemcy)=RFN	148.81.213.6	/pub/info/ntekst.txt
N(Niemcy)=NRD	148.81.213.5	/pub/geografia/n_tekst.html
N(Niemcy)=Saksonia	148.81.213.7	/pub/info/stekst.txt

Definicja 3.25 (zbiór adresów stowarzyszonych z ekstraktem)

Zbiorem adresów stowarzyszonych z ekstraktem EX, nazywamy następujący zbiór:

$$A_{EX} = \bigcup_{t \in EX} A_{t \in EX}$$

■

Przykład

Jeśli wyrażeniem wyszukiwawczym ma być wyrażenie: **w** = Holandia ∧ Niemcy, ekstraktem tego wyrażenia byłby zbiór termów: **EX(W)** = {Holandia, Niemcy}, zaś zbiorem adresów stowarzyszonych z tym ekstraktem mogłyby być tabele 3.7, 3.9 i 3.10

▲

Definicja 3.26 (realizowalność dostępu do sieci)

Dostęp do sieci **S** = **(U, χ)** nazywamy realizowalnym z serwera **u*** dla zlecenia **w** wtedy i tylko wtedy, gdy dla dowolnego adresu **a** $\in A_{EX}$ istnieje ścieżka s_{*-a}.

■

Przykład

Jeśli zleceniem **w** = Holandia ∧ Niemcy (por. poprzedni przykład) a serwerem **u*** będzie serwer – 148.81.213.100, wówczas realizowalność dostępu dla tego serwera oznaczałaby dostępność (połączenie) z wszystkimi serwerami z tabel 3.6 (dla termu Holandia) i 3.7 oraz 3.9 (dla termu Niemcy) z osobna lub przez inne serwery.

▲

Definicja 3.27 (realizowalność obsługi zlecenia)

Proces obsługi zlecenia **w** złożony z (I)ekstrakcji; (II)ewaluacji, (III) dostępu do sieci **S** nazywamy realizowalnym jeśli:

(1) $A_{EX(w)} \neq \varnothing$

(2) dostęp do sieci **S** jest realizowalny

■

Przykład

$A_{EX(w)}$ czyli zbiór adresów stowarzyszonych z ekstraktem wyrażenia **w** nie może być zbiorem pustym oraz serwery o tych adresach są dostępne z serwera **u***. Ilustracją takiej sytuacji jest przedstawiony powyżej przykład z definicji 3.22

Definicja 3.28 (relewantność podsiecl do zlecenia)

Podsiecią relewantną do zlecenia **w** nazywamy sieć **S=(U, χ)** taką, że $\forall u \in U\ \forall t \in EX(w)\ t \in SYG_u$ gdzie $u = \alpha^{-1}(a)$ dla $a \in A_{EX}$; α jest funkcją adresową.

■

Przykład

Tak więc podsiecią relewantną do zlecenia **w**=Holandia ∧ Niemcy będzie zbiór serwerów i połączeń między nimi z tabel 3.7 (dla termu Holandia) i 3.9 oraz 3.10 (dla termu Niemcy) wraz z serwerem dostępu **u*** (czyli – 148.81.213.100).

▲

Twierdzenie 3.4

Jeśli $w = w' \vee w''$ i istnieją podsieci S', S'' relewantne odpowiednio do zleceń **w'**, **w''**, to podsieć relewantna do zlecenia **w** jest również sumą sieci S', S'', tzn. $S' \cup S''$.

Przykład

Niech wyrażeniami będą:

w' = geografia Polski

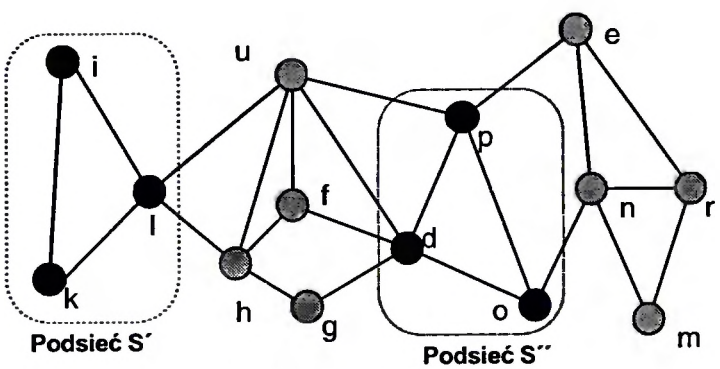
w''= geografia Holandii

$w = w' \vee w''$

Zaś podsieciami relewantnymi będą:

- dla **w'** jest sieć S' składająca się z serwerów: **{i, l, k},**
- dla **w** jest sieć S'' składająca się z serwerów: **{d, p, o}.**

Podsieci S' i S'' przedstawia rysunek poniżej.



Rys. 3.10 Podsieci S', S''

Tabele 3.11 i 3.12 przedstawiają podsieci S′, S″ relewantne odpowiednio do zleceń w′, w″ wraz z nazwą zbioru i wyrażeniami indeksującymi.

Tabela 3.11. Podsieć S′ relewantna do zlecenia w′

Wyrażenie w′	Adres serwera	Nazwa zbioru
geografia Polski	i	tekst1.txt
geografia Polski	l	tekst2.txt
geografia Polski	k	tekst3.txt

Tabela 3.12. Podsieć S″ relewantna do zlecenia w″

Wyrażenie w″	Adres serwera	Nazwa zbioru
Geografia Polski	i	zbiór11.txt
Geografia Polski	k	zbiór12.txt
Geografia Polski	l	zbiór13.txt

Tabela 3.13 przedstawia m.in. zbiór serwerów relewantnych do podanego zlecenia w = w′∨w″, czyli podsieć S′∪ S″.

Tabela 3.13. Suma podsieci S′∪S″ relewantna do zlecenia w = w′∨w″

Wyrażenie w = w′∨w″	Adres serwera	Nazwa zbioru
geografia Polski lub geografia Holandii	i	tekst1.txt
geografia Polski lub geografia Holandii	l	tekst2.txt
geografia Polski lub geografia Holandii	k	tekst3.txt
geografia Polski lub geografia Holandii	p	tekst4.txt
geografia Polski lub geografia Holandii	d	tekst5.txt
geografia Polski lub geografia Holandii	o	tekst6.txt

▲

Fakt ten naturalnie wynika ze sposobu wyznaczania zbioru A_{EX} i odzwierciedla przyjęte tu konserwatywne założenie, dotyczące strategii nawigowania w sieci, że najpierw identyfikowane są wszystkie zasoby, które zawierają termy występujące w wyrażeniu, a dopiero potem wykonywane są operacje logiczne zawarte w wyrażeniu. Założenie to podyktowane jest przeświadczeniem, że wiedza adresatora nie jest pełna. Przykładowo, może się zdarzyć, że dwa termy połączone operatorem ∧ wyznaczają jako relewantne dwa różne zasoby. A zatem należy je odrzucić, gdyż ich przecięcie w świetle wiedzy adresatora jest puste. Jeśli jednak dopuści się, że któryś z adresów jest indeksowany oboma termami (zmiany w sieci zachodzą szybko), to rozsądne jest sprawdzenie, czy fakt ten ma miejsce przed odrzuceniem tych terminów. Naturalnie dane uzyskane z sieci przy powyższym założeniu muszą zostać przygotowane zgodnie z treścią zlecenia (operacje **I, LUB**). Jest to realizowane w fazie opracowywania odpowiedzi (por. rys. 3.1), po czym wynik udostępniany jest zleceniobiorcy.

Obecnie zostanie podane ważne twierdzenie dotyczące realizowalności procesu obsługi zlecenia.

Twierdzenie 3.5

Jeśli sieć **S** jest spójna i istnieje podsieć **S'**, taka że **S'⊆S**, oraz **S'** jest relewantna do zlecenia **w**, to proces obsługi zlecenia **w** jest realizowalny.

Twierdzenie odwrotne nie jest prawdziwe.

Przykład[14]

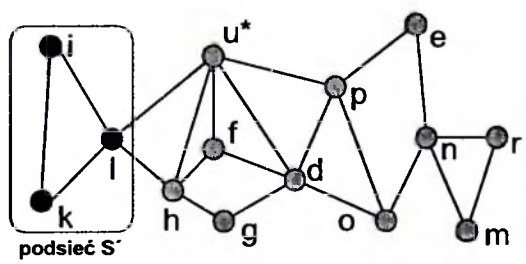
Zdefiniujemy następującą sieć (por. rysunek 3.11):

S – będzie siecią z następującym zbiorem serwerów,
U={i, k, l, u, h, f, d, g, p, n, o, e, r, m} wraz z określoną na nich relacją χ,
S' – podsiecią **(S' ⊆S)** ze zbiorem następujących serwerów **U'={i, k, l}**,
w – wyrażeniem „geografia Polski" realizowalnym przez podsieć **S'** (tabela 3.14.),
u* – serwerem dostępu.

Jak widać z rys. 3.11, sieć jest spójna i istnieje podsieć relewantna do tego zlecenia **w** = geografia Polski (tabela 3.14.), zatem proces obsługi wyrażenia **w** jest realizowalny. Spójność zapewnia istnienie ścieżki dostępu do podsieci **S'** z dowolnego serwera dostępu (w podanym przykładzie z **u***)

Tabela 3.14. Podsieć **S'** relewantna do zlecenia **w**

Wyrażenie **w**	Adres serwera	Nazwa zbioru
Geografia Polski	**i**	zbiór11.txt
Geografia Polski	**k**	zbiór12.txt
Geografia Polski	**l**	zbiór13.txt



Rys. 3.11 Realizowalność obsługi zlecenia w sieci spójnej

Przykładem nieprawdziwości twierdzenia odwrotnego jest rysunek 3.12

Niech:

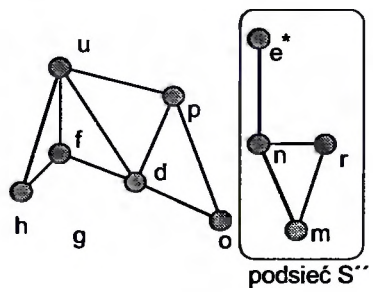
S – będzie siecią z następującym zbiorem serwerów
U={u, h, f, d, g, p, n, o, e, r, m} wraz z określoną na nich relacją χ
S'' – podsiecią **(S''⊆ S)** ze zbiorem następujących serwerów **U''={e, n, r, m}**,

w – wyrażeniem „geografia Francji" realizowalnym przez podsieć S´´ (tabela 3.14.)
e* – serwerem dostępu

Mimo że proces obsługi zlecenia jest realizowalny przez podsieć S´´, to sieć S wcale nie musi być spójna, co pokazuje rysunek 3.12., czyli twierdzenie odwrotne nie jest prawdziwe.

Tabela 3.15. Podsieć S´´ relewantna do zlecenia **w**

Wyrażenie **w**	Adres serwera	Nazwa zbioru
geografia Francji	**e**	zbiór1.txt
geografia Francji	**n**	zbiór2.txt
geografia Francji	**r**	zbiór3.txt
geografia Francji	**m**	zbiór4.txt



Rys. 3.12 Realizowalność obsługi zlecenia w sieci niespójnej

▲

3.5. UCZENIE SIĘ

Proces uczenia[95] się odnosi się w rozważanym modelu do adresatora. Polega on na monotonicznym zwiększaniu wiedzy adresatora o topologii i zasobach sieci. Proces ten przebiega następująco: po zlokalizowaniu wszystkich zasobów sieci, które są relewantne do termów uzyskanych w wyniku ekstrakcji inicjowana jest pętla uczenia się (patrz rys. 3.1.) adresatora. Oto jej kolejne kroki. Każdy term indeksujący relewantne zasoby porównywany jest ze zbiorem termów adresatora. Jeśli adresator zawiera już ten term, to powiększany jest tylko związany z tym ter-

[95] W naukach humanistycznych, zwłaszcza w psychologii termin „uczenie się" ma szeroki i nie do końca zdefiniowany zakres znaczeniowy [MAT79]. Nie jest przedmiotem tej pracy analizowanie tej kwestii. Tutaj „uczenie się" rozumiane jest wąsko i podobnie jak to ma miejsce na gruncie sztucznej inteligencji [CHD84]. Przyjęty w tej pracy zakres znaczeniowy tego terminu został podyktowany względami praktycznymi i ma sens wyjaśniony poniżej.

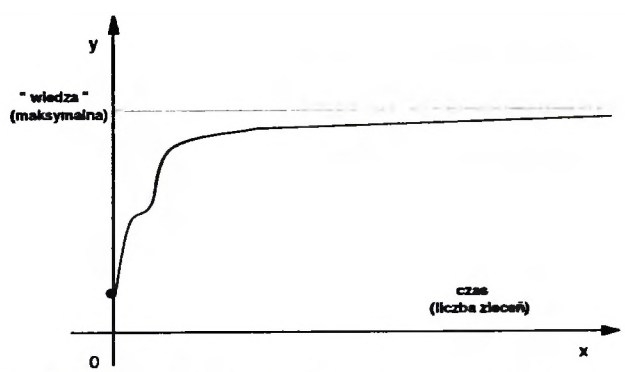
mem zbiór adresów zasobu; w przeciwnym razie do adresatora dodawany jest term i adres. Od razu należy odnotować, że uczenie się nie dotyczy relacji: **S,B,N.** Poniżej zostanie dokładnie przedstawiony proces uczenia się.

- Niech $\mathbf{t} \in \mathbf{T}$ gdzie **T** jest zbiorem termów adresatora **ADR=(T, A, {B,N,S}, τ).**
- Niech $\mathbf{a_i.z_j}$ oznacza adres zasobu $\mathbf{a_i}$, takiego że $\mathbf{u} \in \mathbf{U}$ dla **S=(U, χ)** gdzie **S** jest podsiecią relewantną do **t.**
- Niech $\{\mathbf{a_i.z_1},...,\mathbf{a_i.z_{ni}}\}$ oznacza zbiór wszystkich (adresów) zasobów znajdujących się w serwerze o adresie $\mathbf{a_i}$ relewantnych do pewnego zlecenia.
- Przez $tt_{a_i \cdot z_j}$ oznaczamy listę (ciąg) wszystkich termów indeksujących zasób $\mathbf{a_i.z_j}$.

Dalej będzie wykorzystywana operacja **get_term(tt),** która pobiera pierwszy element z listy **tt.**

Proces uczenia się adresatora opisuje poniższy quasi-Pascalowy program, w którym założono, że podsieć relewantna do zlecenia składa się z **m** serwerów, z których każdy zawiera $\mathbf{n_i}$ zasobów. Napis **NIL** oznacza listę pustą. Należy pamiętać, że A_t^{τ} jest zbiorem adresów stowarzyszonych z termem **t** w adresatorze **ADR.**

```
for i=1,...,m do
for j=1,...,nj do
        while tt_ai·zj ≠NIL do
                begin
                t:=get_term(tt);
                if t ∈ T then A_t^o := A_t^o ∪ {ai.zj}
                        else
                                begin
                                T:=T ∪ {t};
                                τ:= τ ∪ {t, ai.zj}
                                end;
        end while;
end for j;
end for i
```



Rys. 3.13 Przypuszczalny wykres czasu uczenia się systemu

W komentarzu do powyższej procedury należy dodać, krzywa uczenia się adresatora będzie prawdopodobnie zbliżona do krzywej logistycznej, tzn. że po pewnym czasie wiedza systemu zacznie się „nasycać" (ilustruje to rys. 3.13, na którym wiedza systemu odłożona na osi y reprezentuje pewien syntetyczny wskaźnik agregujący liczbę adresów zasobów i termów adresatora. Może to być po prostu suma l_a+l_t, gdzie l_a jest liczbą wszystkich adresów adresatora tzn. $l_a=||A||$, zaś l_t – liczbą wszystkich jego termów tzn. $l_t=||T||$.

Na osi x odłożony jest czas uczenia się mierzony liczbą zleceń skierowanych do systemu. Warto zauważyć, że krzywa uczenia dla „czasu zerowego" wskazuje pewną wartość wskaźnika wiedzy. Odpowiada to wcześniejszemu założeniu, że w chwili uruchomienia systemu adresator dysponuje już pewną wiedzą o sieci. Kształt krzywej uczenia się zależy od typu funkcji używanej do oszacowania wartości progowej ε. Dokładniejsze ustalenie tych zależności wymaga eksperymentu z siecią. Przyjęto też założenie, że indeksowanie zasobów sieci (sygnatur) nie zmienia się, zwiększa się jedynie liczba sygnatur.

4. EKSPERYMENT

4.0. WSTĘP

Trafność i użyteczność przedstawionego w poprzednim rozdziale modelu poddano empirycznej weryfikacji. W tym celu został opracowany system o nazwie NetExp, który stanowi komputerowe odwzorowanie modelu. System zaprojektowano za pomocą obiektowego języka OpenScript systemu ToolBook v.3.0[96] dla środowiska Windows v.3.1. Odnotujmy, że dzięki dynamicznym bibliotekom połączeń (ang. *Dynamic Library Link*) system NetExp ma możliwość komunikacji z pozostałymi aplikacjami środowiska Windows. System składa się z pięciu podstawowych modułów:

- Moduł zadawania pytań, udzielania odpowiedzi i komunikacji,
- Adresator,
- Baza pytania/odpowiedzi,
- Moduł symulacyjnych sygnatur,
- Moduł hipertekstowej pomocy.

W rozdziale tym zostaną omówione: funkcje systemu, jego architektura i problemy, jakie pojawiły się w fazie testowania jego prototypu.

4.1. SYSTEM NETEXP

4.1.1. Baza sprzętowo programowa, język programowania, struktura danych

Poniżej zostaną przedstawione podstawowe pojęcia związane z systemem ToolBook, na bazie którego zbudowano system NetExp. Pojęcia te konieczne są do zrozumienia architektury i działania systemu NetExp.

System ToolBook, język OpenScript, podejście obiektowe

System NetExpt zbudowano na bazie systemu ToolBook. Sam system daje możliwość tworzenia systemów kierowanych zdarzeniami (ang. *event driven*).

[96] Do poprawnej pracy systemu NetExp konieczny jest komputer (co najmniej) z procesorem 386, 4 Mb RAM, oraz ok. 4 MB wolnej przestrzeni na dysku twardym.

ToolBook ze względu na graficzny interfejs projektanta i łatwość programowania jest polecany do tworzenia:

- systemów hipertekstowych/hipermedialnych,
- interaktywnych aplikacji treningowych,
- aplikacji dla baz danych,
- symulacyjnych gier komputerowych.

Dla potrzeb systemu NetExp szczególne zastosowanie znalazła możliwość kierowania zdarzeniami (ang. *event driven*). W programach napisanych w tradycyjnych językach programowania użytkownik wprowadza dane, które są przetwarzane przez program, a ich wyniki przedstawiane użytkownikowi. W programach napisanych na bazie systemu ToolBook, podobnie jak w większości aplikacji środowiska Windows (oraz samym środowisku Windows), każde działanie użytkownika (np.: przyciśnięcie klawisza, wprowadzenie tekstu do pola, wybór okna dialogowego itp.) jest komunikatem, jaki użytkownik wysyła do programu, który dokonuje jego interpretacji. ToolBook tłumaczy każdy komunikat, kierując go do wskazanych[97] obiektów. Kluczowymi obiektami dla systemu ToolBook i dla całego systemu NetExp są tło (strona tylna) oraz strona przednia (strona).

Struktura danych

Podstawowe dane w systemie (struktura adresatora, baza pytań/odpowiedzi) zawarte są w typowych dla systemu ToolBook polach danych (ang. *RecordField*). Każdemu rekordowi odpowiada jedna strona. Oprócz danych adresatora i bazy pytań/odpowiedzi istotną rolę pełnią tzw. symulacyjne sygnatury. Są one reprezentowane przez bazy[98] dBase III. Każde skierowane do systemu pytanie ma charakter transakcji, jego wynikiem jest tworzenie tabel w standardzie dBase III, w których znajdują się odpowiedzi. Dostęp do tych tabel dBase III dokonuje się za pomocą szeregu funkcji z dynamicznej biblioteki połączeń – TB30DB3.DLL. Rolę serwera sygnatur w prototypie systemu NetExp pełni zbiór tekstowy hosty.txt. Komunikacja z tym zbiorem odbywa się przy pomocy dynamicznej biblioteki połączeń – TB30DOS.DLL.

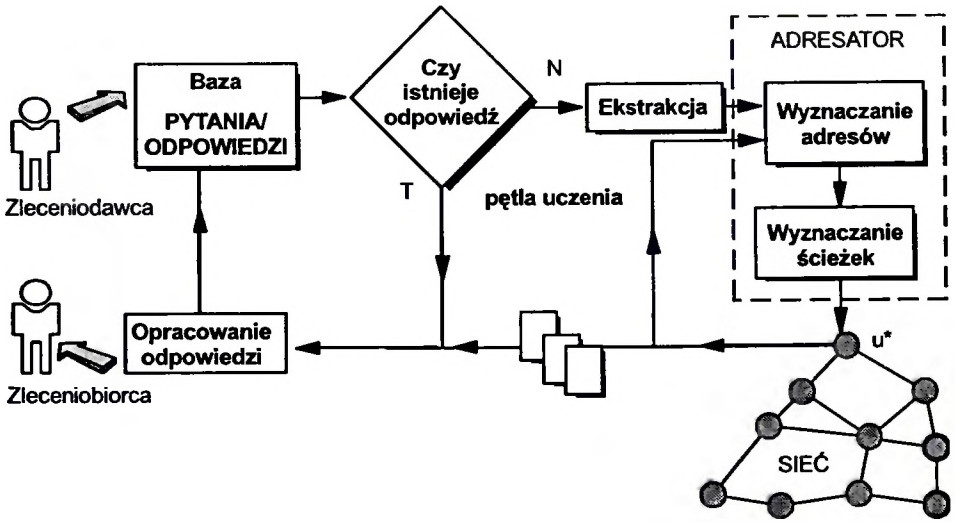
4.1.2. Architektura systemu

Architektura systemu ma strukturę typu klient/serwer. W miarę udoskonalania systemu rola serwera będzie ulegała zmianie. Dla przypomnienia rysunek 4.1 przedstawia model systemu.

---

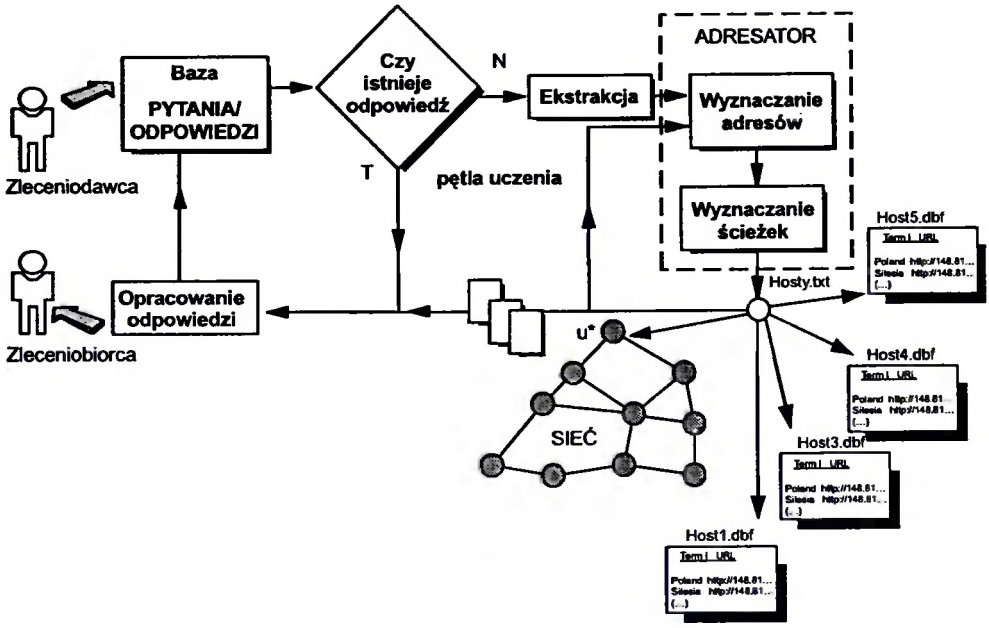
[97] Tzn. zdefiniowanych wcześniej obiektów.

[98] W dalszym ciągu będziemy używać pojęcia tablica na oznaczenie bazy SZBD dBase III.



Rys. 4.1 Architektura klient-serwer w NetExp

Ponieważ nie istnieją jeszcze w sieci Internet sygnatury, dlatego zdecydowano się na zastąpienie ich tzw. symulacyjnymi sygnaturami. Z tego powodu model systemu musiał ulec pewnej zmianie, co ilustruje rysunek poniżej:



Rys. 4.2 Architektura klient-serwer w NetExp z symulacyjnymi sygnaturami

Rolę serwera sygnatur pełni zbiór tekstowy – *hosty.txt* z nazwami wszystkich symulacyjnych sygnatur. W przyszłości planowane jest wprowadzenie sygnatur

do sieci Internet. Obecnie trwają prace nad przygotowaniem sygnatur do współpracy z serwerami HTTPD[99].

Poniżej zostaną przedstawione podstawowe moduły systemu z zaznaczeniem odpowiadających im stron tylnych (teł) w systemie ToolBook.

4.1.3. Moduły w systemie NetExp

System NetExp zbudowano przy wykorzystaniu stron tylnych (teł) i stron przednich systemu ToolBook. Liczba stron tylnych jest ściśle uzależniona od liczby modułów systemu, czyli jego części odpowiedzialnych za funkcjonowanie programu. Liczba stron przednich jest uzależniona od liczby rekordów w danym module i zwiększa się wraz z ilością wiedzy w systemie NetExp. Dzięki przyciskom sterującym umieszczonym w dole ekranu[100] w każdym z modułów użytkownik może przechodzić do dowolnego z modułów. Poniżej zostaną omówione poszczególne moduły programu wraz z przedstawieniem wyglądu stron ekranowych.

Moduł „pytania/odpowiedzi/komunikacja"

Moduł ten zbudowany jest z jednego tła[101] w systemie ToolBook oraz z jednej strony. Dzięki modułowi „pytania/odpowiedzi/komunikacja" użytkownik jest w stanie formułować pytania do systemu. Rysunek 4.3 przedstawia wygląd strony ekranowej dla modułu pytania/odpowiedzi/komunikacja.

Rys. 4.3 Moduł pytania/odpowiedzi/komunikacja

[99] Por. rozdz. 5. *Architektura środowiska współdziałania z siecią.*

[100] Por. rys.: 4.3, 4.4, 4.5, 4.6, 4.8, 4.9.

[101] Nazwa tła w systemie ToolBook – *„Pyt_Odp".*

Prototyp systemu NetExp zakłada na razie istnienie tylko dwóch termów i dwóch rodzajów operatorów Boole'a (*I, LUB*), dlatego użytkownik może formułować pytania składające się z dwóch termów połączonych operatorem Boole'a (*I* lub *LUB*).

Dzięki wydzieleniu osobnych pól na termy proces ekstrakcji przebiega automatycznie już na etapie formułowania pytania. Użytkownik może wybrać dowolny term z pola term_1 lub term_2 albo wpisać samemu term.

Komunikacja z siecią Internet jest możliwa po uzyskaniu odpowiedzi w polu *„Odpowiedzi"*. Użytkownik wybiera kliknięciem myszy dany adres, który pojawia się w polu URL po czym akceptuje operację przesłania żądanego dokumentu klawiszem *„Połączenie z URL przez NETSCAPE"*.

Moduł „baza/pytania/odpowiedzi"

Kolejnym modułem jest „baza/pytania/odpowiedzi", jest on odpowiedzialny za przechowywanie pytań i odpowiedzi oraz komunikację z siecią Internet. Rysunek 4.4 przedstawia wygląd strony ekranowej dla modułu baza/pytania/odpowiedzi.

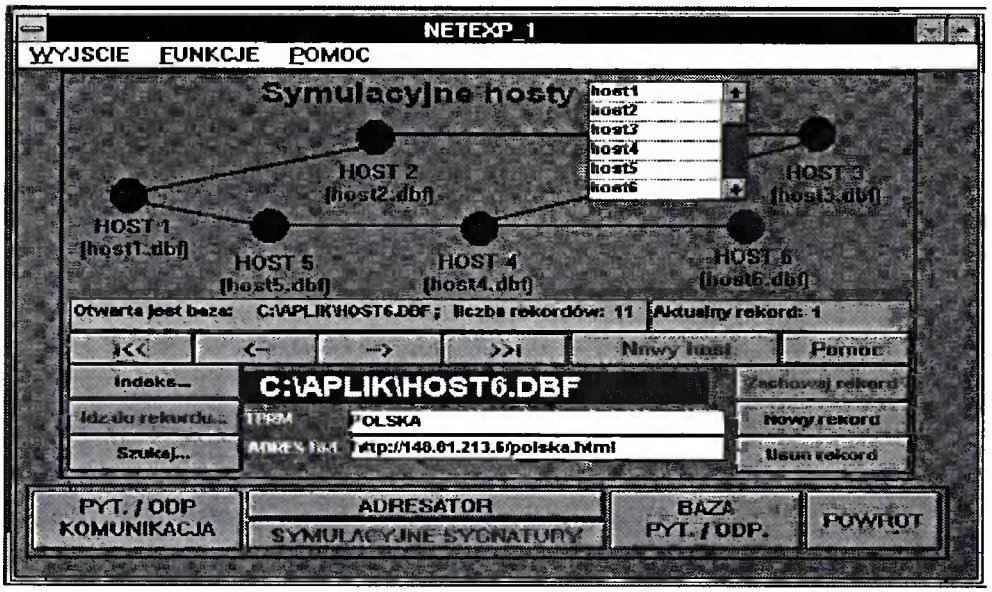
Rys. 4.4 Moduł baza/pytania/odpowiedzi/komunikacja

Do systemu dodawane są tylko te pytania, na które uzyskano odpowiedź, dla której współczynnik dokładności jest równy wartości progowej ε (epsilon). W początkowej fazie istnienia systemu przyjęto, że $\varepsilon = \Theta_w$. Oznacza to, że praktycznie każda odpowiedź systemu jest zapisywana do bazy. Ponadto przyjęto, że o wprowadzeniu rekordu do bazy decydować może sam użytkownik. Z tego modułu możliwa jest również komunikacja z siecią Internet[102].

[102] Patrz rozdział 5.

Moduł quasi-sygnatury

Moduł quasi-sygnatury odpowiada za dodawanie nowych rekordów do sygnatur oraz za wprowadzanie nowych sygnatur (rys. 4.5).



Rys. 4.5 Moduł quasi-sygnatury

Przykładem symulacyjnej sygnatury może być tabela 4.1 (w standardzie dBase III), w której istnieją tylko dwa niepowtarzalne zasoby informacyjne (strony WWW):

http://148.81.213.15/uw/ibin/ibin.htm
http://148.81.213.15/uw/ibin/sss4.htm

Quasi-sygnatury są dwuwymiarowymi tablicami w standardzie dBase III. Każda tablica składa się z dwóch pól typu znakowego, z których jedno jest miejscem na wpisanie termu, a drugie pole służy do wpisania adresu URL. Dzięki zbiorowi hosty.txt system NetExp „wie", do jakich sygnatur ma się odwoływać w czasie procesu wyszukiwawczego. Poniższy rysunek przedstawia wygląd modułu związanego z wprowadzaniem dodatkowych rekordów do symulacyjnych sygnatur oraz dodatkowych sygnatur.

Tabela 4.1 Przykład sygnatury dla komputera 148.81213.15, $Syg_{148.81.213.15.}$

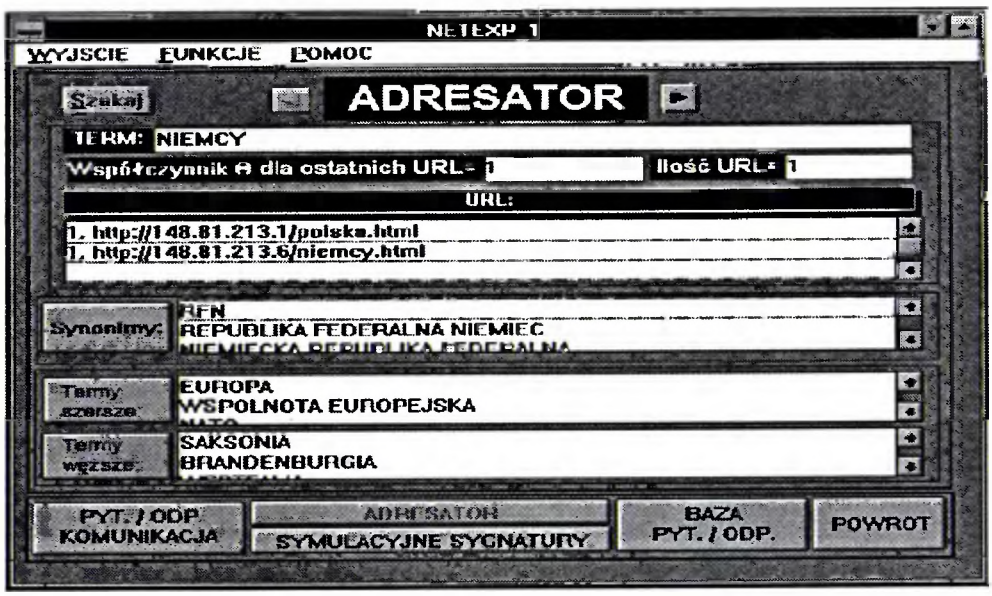
INDEKS	Dokumenty WWW
STUDENT SEMINAR	http://148.81.213.15/uw/ibin/sss4.htm
INFORMATION SCIENCE	http://148.81.213.15/uw/ibin/sss4.htm
INSTITUTE OF LIBRARY AND INFORMATION SCIENCE	http://148.81.213.15/uw/ibin/sss4.htm

UNIVERSITY OF WARSAW	http://148.81.213.15/uw/ibin/sss4.htm
TEMPUS	http://148.81.213.15/uw/ibin/sss4.htm
GERMAN	http://148.81.213.15/uw/ibin/sss4.htm
INSTITUTE OF LIBRARY AND INFORMATION SCIENCE	http://148.81.213.15/uw/ibin/ibin.htm
UNIVERSITY OF WARSAW	http://148.81.213.15/uw/ibin/ibin.htm
CURRICULUM	http://148.81.213.15/uw/ibin/ibin.htm
INSTITUTE OF LIBRARY	http://148.81.213.15/uw/ibin/ibin.htm

Do czasu rozwinięcia rzeczywistych sygnatur, moduł ten będzie pełnił rolę pomocniczą we wprowadzaniu nowych rekordów do wspomnianych tabel.

Adresator

Adresator jest najważniejszą częścią programu NetExp odpowiedzialną za „uczenie się systemu". Rysunek 4.6 przedstawia wygląd strony ekranowej „adresatora".



Rys. 4.6 Adresator

Adresator składa się z jednego tła[103] oraz co najmniej jednej strony. Ilość stron jest uzależniona od ilości termów w adresatorze. Wprowadzanie dodatkowych termów następuje automatycznie w czasie „uczenia się" lub może być dokonywane na życzenie użytkownika. Adresator ma możliwość komunikacji z wy-

[103] Nazwa tła w systemie ToolBook – *„Adresator"*.

branymi adresami URL. Podobnie jak w przypadku bazy pytań/odpowiedzi, do adresatora dodawane są adresy URL, których stopień dokładności równa się wartości progowej ε. Jednak w początkowej fazie eksperymentu przyjęto założenie, że $\varepsilon = \Theta_t$., czyli każdy adres URL dla $\Theta_t > 0$ będzie wprowadzony do adresatora.

Obsługa quasi-tezaurusa w adresatorze

Adresator jest quasi-tezaurusem wraz ze skojarzonymi adresami URL. W tym miejscu należy wytłumaczyć strukturę quasi-tezaurusa w systemie NetExp. Składa się on z poszczególnych „stron", każda strona odpowiada jednemu termowi, oprócz pola *term* znajdują się na stronie pola z termami synonimicznymi, szerszymi oraz węższymi. Wprowadzenie termów synonimicznych, węższych i szerszych należy do użytkownika. Użytkownik może bezpośrednio korzystać z wiedzy adresatora, przeglądając jego termy lub wyszukując wskazane (rys. 4.7). Strony w systemie ToolBook, na bazie którego został zbudowany system NetExp, odgrywają kluczową rolę. Składają się ze strony właściwej oraz tła (strony tylnej). Tło może być elementem wspólnym dla jednej lub wielu stron. W systemie NetExp ilość stron tylnych (teł) jest stała i wynosi pięć stron[104].

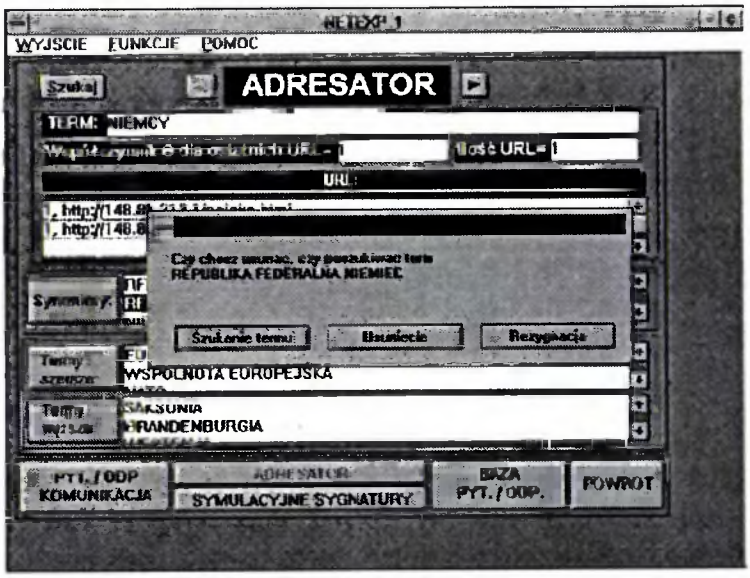
Ilość stron przednich jest zmienna i zależy od wiedzy adresatora i bazy pytań/odpowiedzi może się wahać od 1 do kilkuset stron. W adresatorze istotną rolę odgrywają przesuwne pola danych[105], to w nich zapisywane są adresy URL adekwatne dla danego termu, termy synonimiczne, szersze i węższe. W momencie wprowadzania nowego termu sprawdzana jest wyłącznie jego obecność w polu term, natomiast system nie sprawdza, czy nowy term nie jest przypadkiem synonimem (lub termem szerszym ew. węższym) jakiegoś innego termu. Oprócz tego istnieje możliwość przeprowadzania wyszukiwania danego termu synonimicznego (ew. termu szerszego lub węższego). Po zaznaczeniu urządzeniem wskazującym (myszą) termu w polu *Synonimy* (ew. Termy szersze lub węższe) otwiera się okno dialogowe (rys. 4.7).

Użytkownik może zdecydować się na przeszukiwanie danego termu (synonimu itd.) w adresatorze, jego usunięcie, lub rezygnację z operacji.

---

[104] To na nich znajdują się odpowiednie moduły systemu.

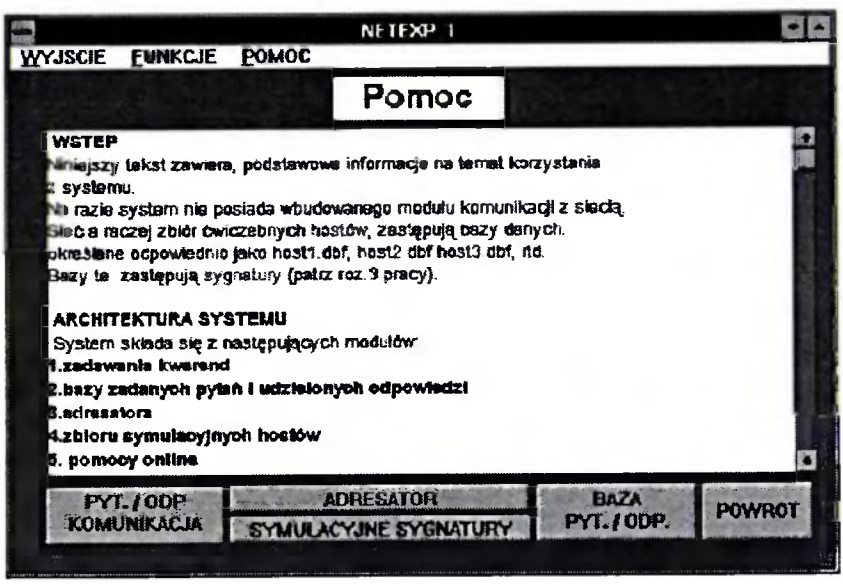
[105] Ang.: *scrolling, single selected listBox, RecordField.*



Rys. 4.7 Poszukiwanie termu skojarzonego z termem głównym

Moduł hipertekstowej pomocy

Rysunek 4.8 przedstawia wygląd strony ekranowej modułu „Pomoc".



Rys. 4.8 Moduł „Pomoc"

Ponieważ system ToolBook przeznaczony jest m.in. do tworzenia aplikacji hipermedialnych, dlatego zdecydowano się na stworzenie pomocy przy wykorzystaniu typowych narzędzi systemu ToolBook.

W miarę udoskonalania systemu NetExp będą podejmowane próby zastąpienia istniejącego modułu pomocy (na bazie systemu ToolBook) standardową pomocą środowiska Windows. Tło *„Pomoc"* pełni również rolę wspomagającą rozwój systemu. Na jednej ze stron tła *„Pomoc"* testowane były komunikaty sterujące (*„handlers"*) zanim zostały włączone jako funkcje do systemu NetExp.

4.2. OBSŁUGA ZLECEŃ SYSTEMU NETEXP

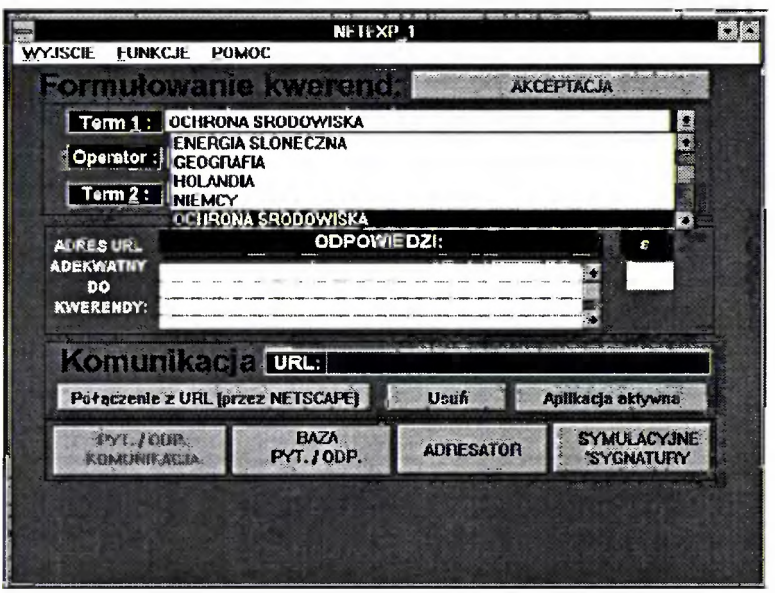
Poniżej zostanie omówione działanie systemu NetExp ze szczególnym uwzględnieniem obsługi zleceń.

4.2.1. Tworzenie pytania

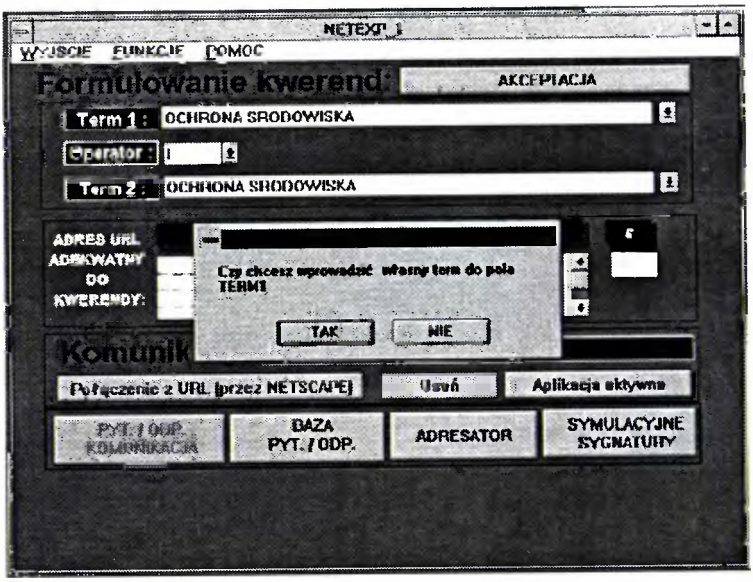
Za formułowanie pytań odpowiedzialny jest moduł tworzenia pytań (rys. 4.9) Formułowanie pytań może odbywać się przez wybranie odpowiedniego termu z samosortującego się okna dialogowego TERM_1 lub TERM_2.

Gdy dany term nie występuje w oknie dialogowym, zleceniodawca po aktywacji pola TERM_1 lub TERM_2 (por. rys. 4.10) może wpisać term wyszukiwawczy.

System zakłada połączenie dwóch termów operatorem Boole'a „I" lub „LUB" przez wybranie odpowiedniego elementu z pola „OPERATOR". Można również zlecić poszukiwanie dla jednego termu, przez aktywacje pola „OPERATOR", co spowoduje zniknięcie samo sortujących się pól „OPERATOR" oraz „TERM_2".



Rys. 4.9 Wybranie termu z samosortującego się pola *„TERM_1"*



Rys. 4.10 Wpisywanie termu do pola

Po sformułowaniu pytania np. *ROSJA I OCHRONA ŚRODOWISKA* użytkownik dokonuje uruchomienia procesu wyszukiwawczo-samouczącego się przez przyciśnięcie myszą klawisza „AKCEPTACJA". System ponownie żąda potwierdzenia pytania i rozpoczyna się proces wyszukiwania.

Procedura sterująca dla klawisza AKCEPTACJA[106] odpowiedzialna jest za obsługę całego zlecenia. Zlecenie składa się z etapów opisanych w poprzednim rozdziale[107], dlatego poniższa procedura obejmuje:

- wyszukiwanie pytania w bazie pytań i odpowiedzi,
- lokalizację zasobów dla termów (pierwszego i drugiego),
- wykonanie operacji logicznych (dla operatora LUB – sumy logicznej znalezionych adresów, dla operatora I – iloczynu logicznego znalezionych adresów).

4.2.2. Przeszukiwanie bazy – pytania/odpowiedzi

Po akceptacji pytania NetExp przechodzi do modułu baza pytania/odpowiedzi[108] i sprawdza czy podobne pytanie nie zostało już wcześniej zadane oraz czy nie uzyskano na nie odpowiedzi, innymi słowy, system sprawdza, czy istnieją adekwatne do niego odpowiedzi w bazie pytań/odpowiedzi (por. rys. 4.11).

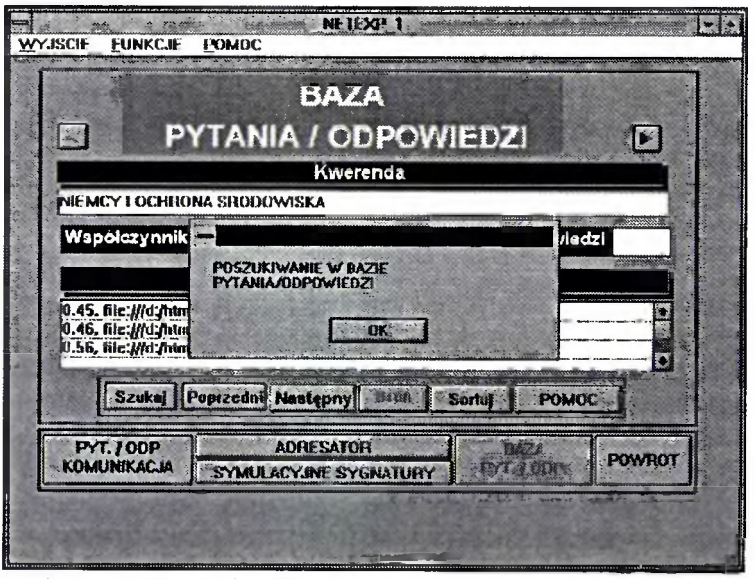
---

[106] W materiałach naukowych biblioteki Instytutu Informacji Naukowej i Studiów Bibliologicznych UW przedstawiony jest program sterujący klawiszem *AKCEPTACJA*.

[107] Por. model systemu w rozdz. 3 i podrozdz.4.1.2.

[108] Nazwa tła w systemie ToolBook – „baza_pyt_odp".

Należy zaznaczyć, że w bazie są przechowywane tylko pytania wraz z odpowiedziami. Jeśli dane pytanie zostało zadane, ale nie uzyskano na nie odpowiedzi lub była ona niezadowalająca (zdaniem systemu lub użytkownika), czyli posiadała zbyt niski współczynnik ε , to nie została ona umieszczona w bazie.



Rys. 4.11 Poszukiwanie pytania w bazie pytania/odpowiedzi

Ze względu na niewielką, początkową wiedzę systemu nie przyjęto w systemie wartości progowej – ε. Jest ona równa współczynnikowi Θ_w dla danego pytania.

4.2.3. Ekstrakcja i poszukiwanie termu w adresatorze

Po stwierdzeniu braku odpowiedzi na zadane pytanie w bazie pytania/odpowiedzi następuje poszukiwanie termów. Pierwszym krokiem jest **ekstrakcja** wyrażenia, czyli wydzielenie jego elementów składowych. I tak np. dla pytania *ROSJA I OCHRONA ŚRODOWISKA* proces ekstrakcji będzie wyglądał następująco:

EX(w)*(ROSJA I OCHRONA ŚRODOWISKA)={ROSJA, OCHRONA ŚRODOWISKA}*

W przypadku systemu NetExp proces ekstrakcji jest zadaniem dość prostym, bowiem ogranicza się zawsze do dwóch termów, które zostały wprowadzone w momencie formułowania pytania. W dalszych planach rozbudowy systemu przyjęto zwiększanie liczby termów.

4.2.4. Przeszukiwanie sygnatur

Szukanie termów składa się z trzech etapów następujących po sobie w zależności od wyników poprzedniego:

Etap I

Etap ten zakłada poszukiwanie termów uzyskanych w wyniku procesu ekstrakcji (*poszukiwanie termów*);

Etap II

Jeśli nie został znaleziony żaden term, wówczas system dokonuje poszukiwania w sygnaturach adresów dla synonimów termu(ów) (*poszukiwanie adresów dla synonimów termu*);

Etap III

Jeśli nie został znaleziony w sygnaturach żaden adres dla synonimów termu, wówczas system poszukuje w sygnaturach adresy dla termów szerszych i węższych.

Symulacyjne sygnatury

Przypomnijmy z rozdz. 3[109] definicję sygnatur jako sumy logicznej zbioru termów dla wszystkich zasobów danego komputera bazowego. Ponieważ opisane w rozdz. 3 sygnatury są na razie postulatem w sieci Internet[110], dlatego postanowiono zastąpić sygnatury zbiorem tzw. *symulacyjnych sygnatur*, które są reprezentowane przez dwuwymiarowe tabele w formacie dBase III. I tak przykładem symulacyjnej sygnatury „host1" może być tabela 4.1. Pola TERM i ADRES są polami znakowymi do 50 znaków.

W fazie eksperymentu zdecydowano się na utworzenie 6 sygnatur. W celu badania wydajności systemu użytkownik może dowolnie zwiększać zbiór symulacyjnych sygnatur (patrz tab. 4.2). Należy jednak przy tym pamiętać o identycznej strukturze tabel oraz o istnieniu zaindeksowanych w niej zasobów. Aby system był w stanie przeszukiwać stworzone przez użytkownika sygnatury należy umieścić (dodać) ich nazwy do istniejącego zbioru SYG\hosty.txt. (patrz tabela 4.3)[111].

Tabela 4.2 Przykład symulacyjnej sygnatury

TERM	ADRES
POLAND	http://148.81.213.5page_11.htm
SILESIA	http://148.81.213.5page_11.htm
POWER STATION	http://148.81.213.5/page_11.htm
CRACOW	http://148.81.213.5/page_11.htm
NOWA HUTA	http://148.81.213.5/page_11.htm
COAL MINE	http://148.81.213.5/page_11.htm
STEELWORKS	http://148.81.213.5/page_11.htm
POLUTION	http://148.81.213.5/htm/tekst.htm
EC	http://148.81.213.5/htm/tekst.htm
NIEMCY	http://148.81.213.5/htm/tekst.htm

[109] Patrz definicja 3.21.

[110] Sieć Internet jest przykładową siecią, w której zdecydowano się przeprowadzić eksperyment.

[111] Por. w podrozdz. 4.1.3. opis modułu symulacyjnych sygnatur.

Tabela 4.3.Zbiór symulacyjnych sygnatur

host1
host2
host3
host4
host5
host6

Przyjęto założenie, że indeksowanie zasobów sieci nie zmienia się, a jedynie zwiększa się ich ilość[112]. W następnym rozdziale[113] przedstawione zostanie zagadnienie automatycznego tworzenia sygnatur dla serwerów WWW.

Szukanie termów w sygnaturach

Szukanie adresów dla termów synonimicznych oraz szerszych i węższych w sygnaturach zależne jest od ilości termów i powiązanych z nimi termów synonimicznych, szerszych i węższych w adresatorze.

W przypadku, gdy term nie zostanie znaleziony w adresatorze, system automatycznie dodaje term do adresatora i uruchamia funkcję *szukajTermAdrAllSyg (v_term, v_baza, PageRef)*, która przeszukuje wszystkie dostępne sygnatury. Parametrami funkcji są:

- *v_term, v_baza, PageRef,*
- *v_term* – nazwa poszukiwanego termu,
- *v_baza* – baza, w której będą zapisywane wyniki,
- *PageRef* – lokalizacja termu w sygnaturze.

Rysunek 4.12 przedstawia wygląd strony ekranowej w momencie działania opisanej funkcji. Należy zaznaczyć, że poszukiwanie termu dla każdej sygnatury jest sygnalizowane komunikatem (patrz. rys. 4.12). Można założyć, że czas wyszukiwania staje się przez to bardziej przybliżony do przeszukiwania rzeczywistych sygnatur w rozległych sieciach komputerowych.

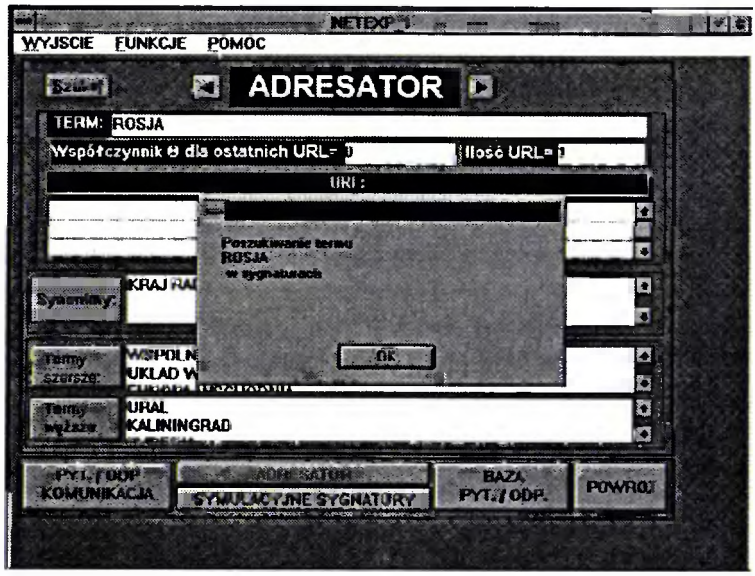
Funkcja *szukajTermAdrAllSyg (v_term, v_baza, PageRef)*, jest zastosowaniem bardziej szczegółowej funkcji *szukajTermAdrSyg (v_term, v_baza, v_host, PageRef)*, w której oprócz wspomnianych parametrów wprowadzona jest nazwa sygnatury, do której ma być skierowane zlecenie.

Wspomniane poniżej funkcje: *szukajTermAdrAllSyg (), szukajSynAllSyg ()* oraz *szukajBNAllSyg ()* przeszukują wszystkie sygnatury, odwołując się do przedstawionego wcześniej tekstowego zbioru hosty.txt.

[112] Co oznacza, że struktura tabel (symulacyjnych sygnatur) wraz z nazwami pól pozostaje taka sama.

[113] Patrz rozdz. 5: *Współdziałanie systemu NetExp z siecią.*

Funkcja *szukajTermAdrSyg (v_term, v_baza, v_host, PageRef)* poszukuje w danej symulacyjnej sygnaturze adresy wskazanego termu[114]. W obiektowym języku OpenScript funkcja ta działa dzięki wspomnianym wcześniej dynamicznym bibliotekom połączeń (ang. *Dynamic Library Link*) – TB30DB3.DLL. Efektem działania funkcji *szukajTermAdrSyg ()* jest utworzenie tablicy w standardzie dBase III i zapisanie do niej wyników wyszukiwania. Funkcja ta zwraca, w przypadku gdy NetExp nie znajdzie żadnego adresu, *fałsz* (ang. *FALSE*). Zmienna s_bookPath pozwala odnajdywać tworzone tablice i zbiory na dysku lokalnym.



Rys. 4.12 Poszukiwanie termu w sygnaturach

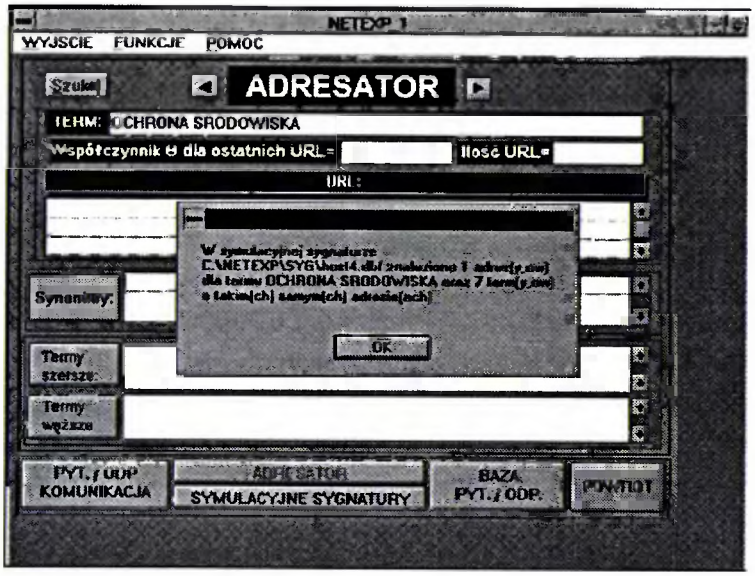
Najpierw system tworzy pustą dwuwymiarową tablicę – *tempBaza.* W rzeczywistości *tempBaza* jest nazwą zmiennej, w której przechowywana jest niepowtarzalna nazwa tablicy dBase III[115]. Do niej wpisywane są wszystkie adresy relewantne do szukanego termu. Następnie dodawane są wyniki z bazy tymczasowej (*temp*). Po przeszukaniu wszystkich adresów relewantnych do szukanego termu rozpoczyna się proces dodawania wszystkich adresów z bazy tymczasowej (*tempBaza*) oraz wszystkich termów danej sygnatury do bazy wynikowej (*temp1.dbf*). Na powrót przeszukiwana jest dana sygnatura. Tym razem kluczem wyszukiwawczym nie jest szukany term, lecz zbiór znalezionych wcześniej adresów i zapisanych do bazy *tempBaza.* Proces wyszukiwania powtarzany jest dla każdej sygnatury, oczywiście według danych ze zbioru *hosty.txt* (symulacyjnego „serwera" symulacyjnych sygnatur).

[114] Patrz niżej.

[115] Nazwą tworzonej tymczasowo bazy jest *384793.dbf.*

Wyniki poszukiwań dla termu pierwszego zapisywane są w bazie *temp1.dbf*, a dla termu drugiego w bazie *temp2.dbf*. Rysunek 4.13 przedstawia zakończenie procesu odnajdywania adresów dla jednego z termów oraz wszystkich termów o takich samych adresach, jak znaleziony wcześniej term. Operacje te powtarzane są dla wszystkich znanych sygnatur dzięki funkcji *szukajTermAdrAllSyg (v_term, v_baza, PageRef)*.

Po pozytywnym zakończeniu poszukiwań w sygnaturach utworzona jest baza, która jest punktem wyjścia dla procesu uczenia się adresatora. Tabela 4.4 przedstawia przykładową bazę. Jak widać tabela 4.4 posiada oprócz znalezionego termu *„OCHRONA ŚRODOWISKA"* i skojarzonych z nim adresów, również pozostałe termy o takich samych adresach *„ENERGIA SŁONECZNA, MAZOWSZE, NIEMCY"* itp. Ich istnienie jest warunkiem koniecznym do zaistnienia w procesie uczenia się systemu.



Rys. 4.13 Pomyślne zakończenie procesu przeszukiwania danej sygnatury

Po zakończeniu procesu przeszukania wszystkich sygnatur uruchamiane są dwie funkcje *uczenieAdresatora (v_baza)*, odpowiedzialne za uczenie adresatora (por. rys. niżej) oraz *usunZbedneTermy (v_baza,v_term)*. Celem tej ostatniej funkcji jest usunięcie wszystkich powtarzających się adresów URL. W ostatniej fazie działania systemu zbiór będący wynikiem działania funkcji *usunZbedneTermy (v_baza,v_term)* jest niezbędny przy tworzeniu operacji sumy lub iloczynu logicznego znalezionych adresów, czyli operacji: „I" lub „LUB".

Tablica 4.4 Tablica *„temp2.dbf"* dla termu *„OCHRONA ŚRODOWISKA"*

TERM_TEMP	ADRES_TEMP
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_2.htm
POLSKA	http://148.81.213.3/info_2.htm
MAZOWSZE	http://148.81.213.3/info_2.htm
ŚLĄSK	http://148.81.213.3/info_2.htm
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_4.htm
SAKSONIA	http://148.81.213.3/info_4.htm
NIEMCY	http://148.81.213.3/info_4.htm
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_5.htm
ENERGIA SŁONECZNA	http://148.81.213.3/info_5.htm
FUNDACJE	http://148.81.213.3/info_5.htm
OCHRONA ŚRODOWISKA	http://148.81.213.5/info_6.htm
POLSKA	http://148.81.213.5/info_6.htm
NIEMCY	http://148.81.213.5/info_6.htm
MAZOWSZE	http://148.81.213.5/info_6.htm
ENERGIA SŁONECZNA	http://148.81.213.5/info_6.htm
ZANIECZYSZCZENIA	http://148.81.213.5/info_6.htm
ELEKTROWNIE WODNE	http://148.81.213.5/info_6.htm
POMORZE	http://148.81.213.5/info_6.htm

Szukanie adresów dla termów synonimicznych

Gdy system nie znajduje w sygnaturach żadnych adresów dla danego termu[116], następuje próba znalezienia wszystkich adresów dla termów synonimicznych (rys. 4.14). Wyszukiwaniem steruje funkcja *szukajSynAllSyg (v_term, v_baza, PageRef)*, w przypadku znalezienia co najmniej jednego adresu zwraca ona liczbę znalezionych adresów oraz współczynnik aproksymacji termu – Θ, który autor arbitralnie przyjął jako 1, czyli tak samo jak dla termów. Adresy te jednak są rozróżnialne przez użytkownika, mają dodatkowy parametr w adresatorze – literę „S".

Szukanie adresów dla termów szerszych i węższych

Gdy nie zostaną znalezione żadne z adresów dla termów synonimicznych rozpoczyna się proces poszukiwania termów szerszych i węższych względem danego, o ile takie istnieją w adresatorze (patrz rys. 4.15).

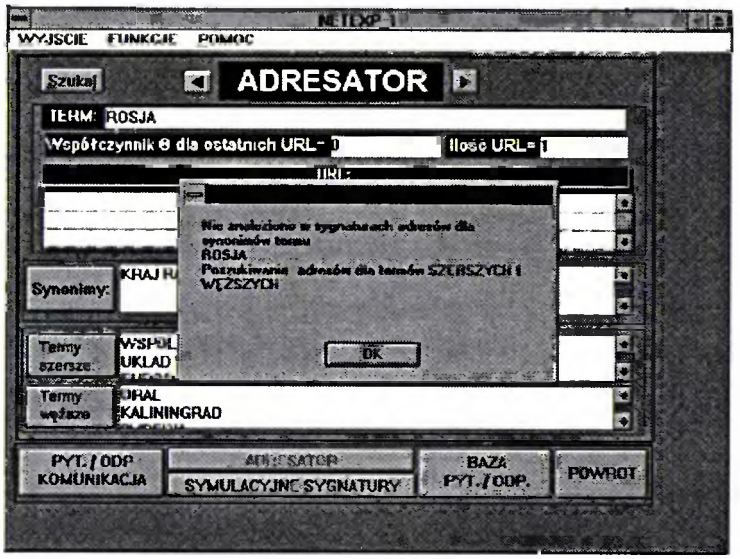
Dzięki funkcji *szukajBNAllSyg (v_term, v_baza, PageRef)* przeszukiwane są wszystkie sygnatury w poszukiwaniu adresów dla termów szerszych i węższych. Działanie funkcji *szukajBNAllSyg (v_term, v_baza, PageRef)*[117] jest zastosowaniem szczegółowej funkcji – *szukajBNSyg (v_term,PageRef,v_host,v_baza)*, prze-

[116] Czyli wartością funkcji *szukajTermAdrSyg ()* jest „FALSE".

[117] Analogiczne do funkcji *szukajSynAllSyg(v_term, v_baza, PageRef)*.

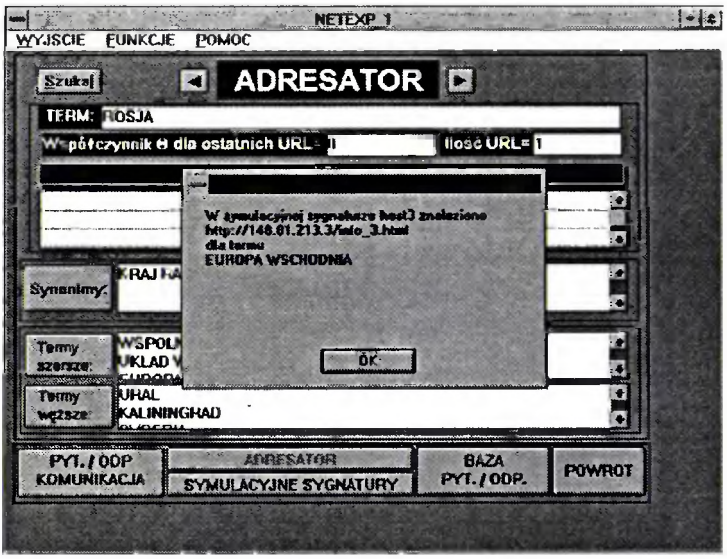
szukującej pojedynczą sygnaturę. W przypadku znalezienia termu (ew. synonimicznego, szerszego lub węższego) użytkownik jest o tym poinformowany (patrz rys. 4.16).

Rys. 4.14 Poszukiwanie adresów synonimów termu w sygnaturach



Rys. 4.15 Poszukiwanie adresów dla termów szerszych i węższych

Po przeszukaniu wszystkich sygnatur w przypadku, gdy adresy dla termów szerszych lub węższych (synonimicznych) zostały znalezione w sygnaturze, tworzony jest zbiór adresów.



Rys. 4.16 Znalezienie termu w sygnaturze

4.2.5. Współczynnik aproksymacji termów

Współczynnik aproksymacji termów Θ_t dotyczy: samych termów, synonimów, wyrażeń szerszych i węższych. Dla termów oraz wyrażeń synonimicznych współczynnik aproksymacji wynosi 1. Podana w rozdziale 3 i 4 funkcja stosuje się do termów szerszych i węższych.

Przypomnijmy, że dla współczynnika aproksymacji termu Θ_t przyjęto następujące założenia:

1. $0 \le \Theta_t \le 1$
2. $\lim_{l \to \infty} \Theta_t = 1$

   l jest liczbą wszystkich adresów URL znalezionych dla termów szerszych i węższych względem danego oraz

   $$l = || A_B^l \cup \ldots \cup A_B^n \cup A_N^l \cup \ldots \cup A_N^k ||$$

3. $\Theta_t \big|_{l=0} = 0$

Przypomnijmy, że w rozdz. 3 zdecydowano się na przyjęcie następującej funkcji:

$$\Theta_t = \frac{\sqrt{(l-1)^2} - 1}{l-1}$$

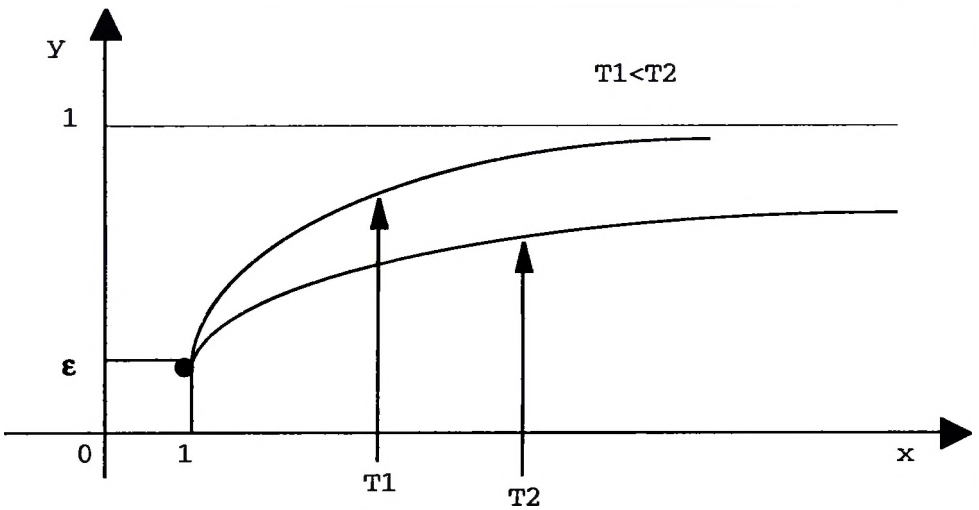
Jednak w czasie eksperymentu okazało się, że wartości tej funkcji dla stosunkowo małych l (2, 3, 5) wynosiły powyżej 0.8, dlatego została przyjęta inna funkcja:

$$\Theta_l = 1 - \left[(1-\varepsilon)_e^{\frac{1-l}{100}} \right]$$

gdzie:

l – liczba URL
ε – **0.1,** dla i=1, f(i)=0.123.
T – **100,** ten czynnik jest odpowiedzialny za spłaszczenie funkcji.
e – exp

Poniżej przedstawiono dwa wykresy nowej funkcji $f(l) = 1 - \left[(1-\varepsilon)_e^{\frac{1-l}{100}} \right]$ dla dwóch różnych współczynników T (T1<T2.).



Rys. 4.17 Dwa wykresy nowej funkcji $\Theta_l = 1 - [(1-\varepsilon)e^{\frac{1-l}{100}}]$, dla dwóch różnych T (T_1<T_2)

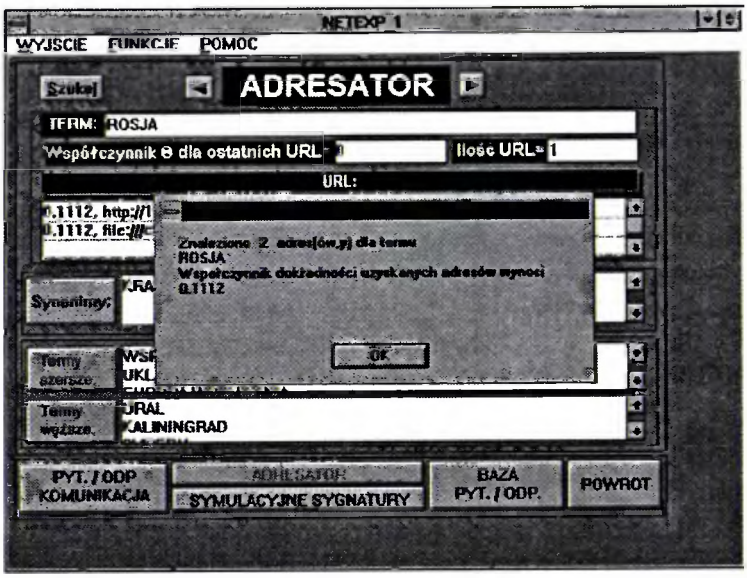
Poniżej zostaną przedstawione przykładowe wartości funkcji dla różnych *l* dla takiego samego współczynnika spłaszczenia funkcji T=100.

f(1)=0,1000
f(2)=0,1112
f(5)=0,1439
f(10)=0,1958
f(20)=0,2903
f(30)=0,3737
f(40)=0,4473
f(50)=0,5122
f(60)=0,5695
f(70)=0,6201
f(80)=0,6647
f(90)=0,7041
f(100)=0,7389

Ponieważ system ToolBook nie ma możliwości obliczania funkcji exp, dlatego podana funkcja została obliczona poza systemem, a wartości i argumenty funkcji zachowane w bazie danych (*x100.dbf*).

Po wyznaczeniu wartości współczynnika aproksymacji zlecenia użytkownik może podjąć decyzję, czy warto kierować dane zlecenie do sieci. W tej sprawie proponuje się arbitralne ustalenie wartości progowej $0<\varepsilon\leq 1$.

Ze względu na niewielką wiedzę systemu nie przyjęto wartości progowej, a o skierowaniu zlecenia do sieci decyduje sam użytkownik. Po wzbogaceniu się wiedzy adresatora system podaje współczynnik aproksymacji termu (por. rysunek 4.18).



Rys. 4.18 Podanie współczynnika aproksymacji termu

4.2.6. Budowanie odpowiedzi

Usuwanie powtarzających się adresów

Załóżmy, że pytanie brzmiało:

„POLAND AND ENVIRONMENTAL LEGISLATION".

Tabele 4.5 i 4.6 przedstawiają znalezione adresy dla termów

POLAND
ENVIRONMENTAL LEGISLATION.

Θ*term1* oznacza współczynnik aproksymacji pierwszego termu

Θ*term2* oznacza współczynnik aproksymacji drugiego termu

$\Theta_{term1} = \Theta_{„POLAND"} = 0{,}1112$

$\Theta_{term2} = \Theta_{„ENVIRONMENTAL\ LEGISLATION"} = 1$

Tabela 4.5 Baza *„temp1.dbf"* dla termu *„POLAND"*

TERM_TEMP	ADRES_TEMP
EASTERN EUROPE	http://148.81.213.3/info_4.htm
WARSAW PACT	http://148.81.213.3/info_2.htm
SILESIA	http://148.81.213.3/info_2.htm
WARSAW	http://148.81.213.3/info_2.htm
MAZOVIA	http://148.81.213.3/info_4.htm

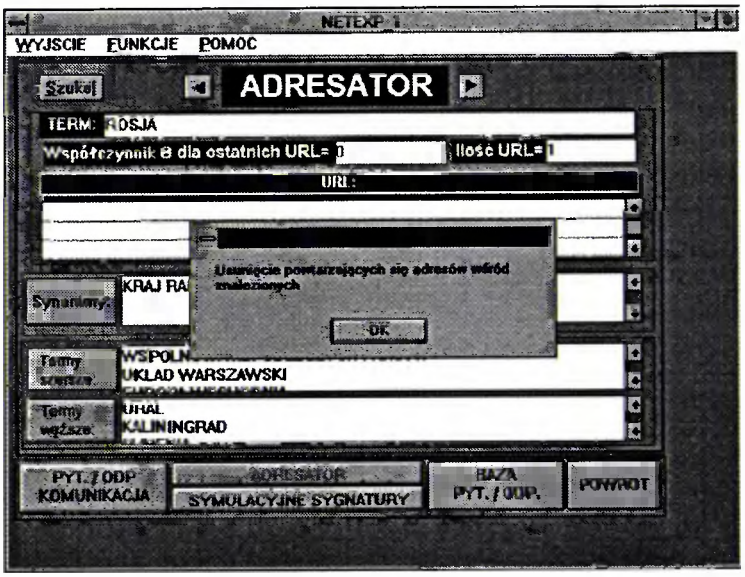
Przypomnijmy, że wynikiem działania funkcji *szukajTermSyg ()* było utworzenie dwóch tabel. W pierwszej (*tempBaza*) wprowadzano adresy dla danego termu, następnie do tabeli wynikowej[118] wprowadzono wszystkie termy, które miały takie same adresy jak adresy z tabeli *tempBaza*. W efekcie powstała tabela z powtarzającymi się adresami. Do obliczenia współczynnika aproksymacji termów oraz późniejszych operacji logicznych konieczne było usunięcie nadmiarowych rekordów. W tym celu stworzono funkcje *usunPowtRekordy ()*, która jest wywoływana jako podprogram w funkcji odpowiedzialnej za zwiększanie się wiedzy adresatora *wiedzaAdresatora ()*. Usunięcie powtarzających się rekordów sygnalizowane jest przez system (rys. 4.19).

Tabela 4.6 Baza *„temp2.dbf"* dla termu *„ENVIRONMENTAL LEGISLATION"*

TERM_TEMP	ADRES_TEMP
ENVIRONMENTAL LEGISLATION	http://148.81.213.3/info_2.htm
POLSKA	http://148.81.213.3/info_2.htm
MAZOWSZE	http://148.81.213.3/info_2.htm
SLASK	http://148.81.213.3/info_2.htm
ENVIRONMENTAL LEGISLATION	http://148.81.213.3/info_4.htm
SAKSONIA	http://148.81.213.3/info_4.htm
NIEMCY	http://148.81.213.3/info_4.htm
ENVIRONMENTAL LEGISLATION	http://148.81.213.3/info_5.htm
SOLAR ENERGY	http://148.81.213.3/info_5.htm
FUNDACJE	http://148.81.213.3/info_5.htm
ENVIRONMENTAL LEGISLATION	http://148.81.213.5/info_6.htm
POLSKA	http://148.81.213.5/info_6.htm
NIEMCY	http://148.81.213.5/info_6.htm

[118] Dla termu pierwszego tabeli – *temp1.dbf*, dla termu drugiego tabeli – *temp2.dbf*.

MAZOWSZE	http://148.81.213.5/info_6.htm
SOLAR ENERGY	http://148.81.213.5/info_6.htm
ZANIECZYSZCZENIA	http://148.81.213.5/info_6.htm
ELEKTROWNIE WODNE	http://148.81.213.5/info_6.htm
POMORZE	http://148.81.213.5/info_6.htm



Rys. 4.19 Usunięcie zbędnych adresów

Obliczanie współczynnika odpowiedzi

Przypomnijmy, że współczynnik aproksymacji zlecenia wynosił[119]:

$$\Theta_w = \frac{\sum_{i=1}^{n} \Theta_{t_i}}{n}$$

W systemie NetExp w module PYTANIA/ODPOWIEDZI, jest możliwość wprowadzania tylko dwóch termów, dlatego współczynnik aproksymacji odpowiedzi wygląda następująco:

$$\Theta_w = \frac{\Theta_{term\ 1} + \Theta_{term\ 2}}{2}$$

Dla podanych przykładów współczynniki aproksymacji termów wynosi:

$$\Theta_{term1} = \Theta_{„POLAND"} = 0{,}1112,$$

$$\Theta_{term2} = \Theta_{„ENVIRONMENTAL\ LEGISLATION"} = 1$$

[119] Por. rozdz. 3.

Zatem współczynnik aproksymacji całej odpowiedzi wynosi:

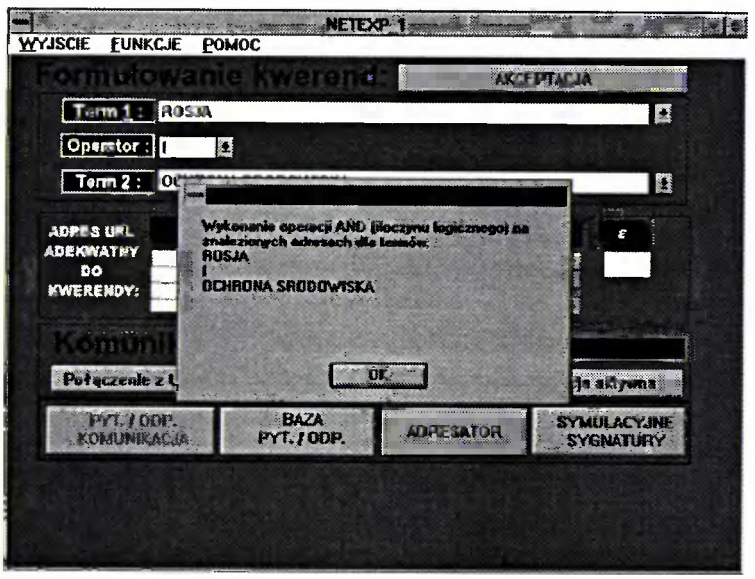
$$\Theta_w = \frac{1+0{,}1112}{2} = 0.5556$$

Obliczenie współczynnika odpowiedzi następuje dopiero po zakończeniu procesu wzbogacania wiedzy adresatora.

Wykonanie operacji logicznych na zbiorach znalezionych adresów.

Po zakończeniu procesu lokalizacji zasobów dla danych termów (por. proces uczenia się i zwiększanie wiedzy systemu) następuje wykonanie operacji iloczynu logicznego lub sumy logicznej[120] (rys. 4.20).

Przypomnijmy, że efektem działania wspomnianej funkcji *wiedzaAdresatora (v_baza)* jest utworzenie dwuwymiarowej relacyjnej bazy danych o polach typu znakowego: *termTemp:* reprezentujących szukany term (np. *OCHRONA ŚRODOWISKA*) oraz pola *adresTemp,* zawierających zbiór adresów adekwatnych dla szukanego termu. Po usunięciu powtarzających się adresów dla obu termów utworzone zostały dwie tablice: *„temp1.dbf"* i *„temp2.dbf"* (patrz tabela 4.7 i 4.8).



Rys. 4.20 Komunikat wykonywania operacji iloczynu logicznego (operator *„I"*)

Tabela 4.7. Tablica *„temp1.dbf"* dla termu *„ROSJA"*

TERM_TEMP	ADRES_TEMP
ROSJA	http://148.81.213.3/info_3.htm
ROSJA	http://148.81.213.3/info_4.htm

[120] Czyli operacja „I" lub „LUB".

Tabela 4.8. Tablica *„temp2.dbf"* dla termu *„OCHRONA ŚRODOWISKA"*

TERM_TEMP	ADRES_TEMP
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_2.htm
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_4.htm
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_5.htm
OCHRONA ŚRODOWISKA	http://148.81.213.5/info_6.htm

Wraz z utworzeniem tabel *„temp1.dbf"* i *„temp2.dbf"* znane są wyniki współczynnika aproksymacji zlecenia Θ_w. Jak już wcześniej zaznaczono efektem działania operacji *wiedzaAdresatora ()* jest, oprócz stworzenia bazy wynikowej (np. *„temp1.dbf"*), podanie współczynnika dokładności Θ_t.

Następnie system dokonuje operacji logicznych[121] na zbiorach adresów z tablicy *„temp1.dbf"* i *„temp2.dbf"*, a wynikiem tej operacji jest wybranie adresów wspólnych dla obu tabeli, czyli utworzenie tabeli *„temp3.dbf"* (patrz tabela 4.9).

Tabela 4.9. Tablica *„temp3dbf"* dla odpowiedzi *„ROSJA I OCHRONA ŚRODOWISKA"*

TERM_TEMP	ADRES_TEMP
ROSJA I OCHRONA ŚRODOWISKA	http://148.81.213.3/info_3.htm
ROSJA I OCHRONA ŚRODOWISKA	http://148.81.213.3/info_4.htm

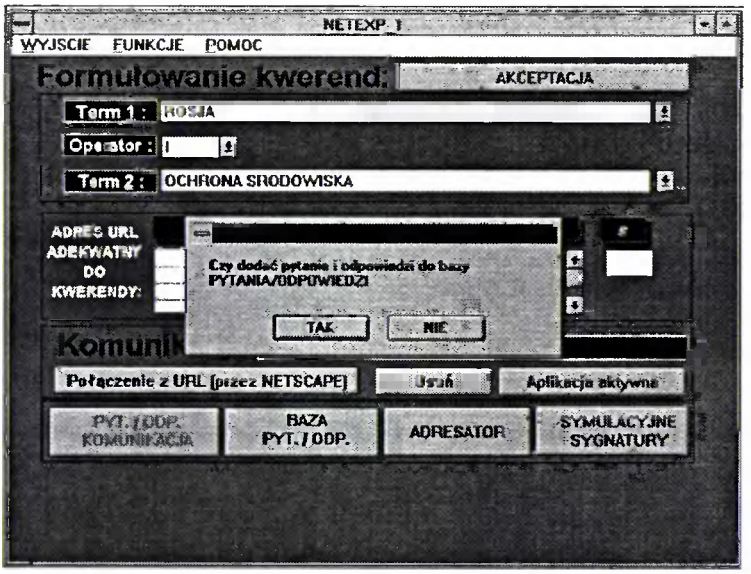
W przypadku gdyby termy były rozdzielone operatorem *LUB* (suma logiczna adresów dla termów), to baza wynikowa wyglądałaby jak w tabeli 4.10.

Tabela 4.10. Tablica *„temp3.dbf"* dla odpowiedzi *„ROSJA LUB OCHRONA ŚRODOWISKA"*

TERM_TEMP	ADRES_TEMP
ROSJA LUB OCHRONA ŚRODOWISKA	http://148.81.213.3/info_3.htm
ROSJA LUB OCHRONA ŚRODOWISKA	http://148.81.213.3/info_2.htm
ROSJA LUB OCHRONA ŚRODOWISKA	http://148.81.213.3/info_4.htm
ROSJA LUB OCHRONA ŚRODOWISKA	http://148.81.213.3/info_5.htm
ROSJA LUB OCHRONA ŚRODOWISKA	http://148.81.213.5/info_6.htm

Wprowadzenie pytania i znalezionej odpowiedzi jest poprzedzone pytaniem systemu (rys. 4.21). Należy bowiem pamiętać, że zmiany w sieci zachodzą dość szybko, dlatego uzyskane przez system odpowiedzi na zadane pytanie mogą być po jakimś czasie dla użytkownika nieaktualne.

[121] Suma – „LUB", iloczyn – „I".



Rys. 4.21 Potwierdzenie wprowadzenia do bazy pytania i odpowiedzi

W systemie przyjęto monotoniczny model uczenia się systemu. Co oznacza, że każde następne odpowiedzi na pytanie nie zmniejszają wiedzy systemu.

Po zakończeniu całej operacji logicznej („I" lub „LUB") użytkownik ma do dyspozycji zbiór odpowiedzi URL (rys. 4.22). Pierwszy element (rys. 4.22) przedstawia współczynnik odpowiedzi Θ_w, następny przedstawia adres URL, z którym użytkownik może połączyć się i uzyskać dokument. Następny rozdział[122] omawia mechanizm łączenia się z wybranymi przez NetExp zasobami sieci.

Rys. 4. 22 Uzyskanie odpowiedzi adekwatnej do zlecenia

[122] rozdz. 5. *Architektura środowiska współdziałania z siecią.*

4.3. ADRESATOR – INTELIGENTNY QUASI-TEZAURUS

Adresator, mimo że stanowi integralną część systemu NetExp, wymaga osobnego omówienia głównie ze względu na samouczący się mechanizm dostępu do sieci (por. rozdz. 3). Zostanie poniżej omówiona struktura adresatora, mechanizm zdobywania wiedzy i mechanizm uczenia się adresatora.

4.3.1. Struktura quasi-tezaurusa

Przypomnijmy z rozdziału 3 strukturę adresatora. *ADRESATOR* został zdefiniowany jako czwórka uporządkowana:

$$ADR=(T, A, \{B,N,S\}, \tau), \text{ gdzie:}$$

T – jest zbiorem termów; $T \neq \emptyset$;

A – jest zbiorem adresów serwerów sieci (ewentualnie rozszerzonych nazwami zasobów, konwencja kropkowa", def. 3.4);

{B,N,S} – są relacjami „szersze", „węższe", „synonimii" określonymi wyżej w zbiorze **T**;

τ – jest relacją adresującą $\tau \subseteq T \times 2^A$ mającą własność $\forall\, t \in T\ \forall A' \subseteq A\ \forall\, a \in A'\ (u\ \tau\ A' \rightarrow t \in SYG_{\alpha^{-1}(a)})$, gdzie α jest funkcją adresową (def. 3.3.).

Adresator jest zatem zbiorem termów, na którym określono relację i ponadto każdemu termowi **t** przyporządkowano zbiór adresów tych serwerów sieci, których sygnatury zawierają ten term.

Tabela 4.11

TERM	URL address
Szwecja	http://148.81.213.81/pub/geografia/pltekst.htm
Dania	http://148.81.213.16/pub/info/pltekst.txt
Holandia	http://148.81.213.21/pub/geografia/pltekst.htm
Francja	http://148.81.213.4/pub/kultura/doc.htm
S(Rzeczypospolita)=Poland	http://148.81.213.6/pub/info/pltekst.txt
S(Stany Zjednoczone)=USA	http://148.81.213.5/pub/geografia/pltekst.htm
S(Niemcy)=RFN	http://148.81.213.7/pub/info/pltekst.txt
B(Kanada)=Ameryka Północna	http://148.81.213.16/pub/info/pltekst.txt
B(Korea)=Azja	http://148.81.213.21/pub/geografia/pltekst.htm
B(Polska)=Europa Wschodnia	http://148.81.213.10/pub/kult/pltekst.htm
N(Stany Zjednoczone)=Kalifornia	http://148.81.213.8/pub/geografia/pltekst.htm
N(Polska)=Mazowsze	http://148.81.213.52/pub/geografia/pltekst.htm

Wyjaśnienie do tabeli 4.11:

S(Polska) oznacza synonim termu „Polska"

B(Polska) oznacza term szerszy względem termu „Polska"

N(Polska) oznacza term węższy względem termu „Polska"

Każdy element adresatora jest więc parą: term-zbiór adresów serwerów, gdzie znajdują się zasoby indeksowane tym termem. Pewną ilustracją adresatora może być tabela 4.11.

4.3.2. Wiedza adresatora

W systemie NetExp uczenie się adresatora nie dotyczyło termów synonimicznych, szerszych ani węższych. Dla sprawnego działania systemu przyjęto, że funkcja odpowiedzialna za uczenie się adresatora – *uczenieAdresatora ()* jest częścią większej funkcji *wiedzaAdresatora ()*. W dalszych rozważaniach związanych z adresatorem opisane zostaną operacje związane z tzw. *zwiększaniem wiedzy adresatora*[123] bez procesu uczenia się, następnie omówione zostanie uczenie się adresatora.

Zwiększanie wiedzy systemu

Zwiększanie wiedzy systemu oznacza uczenie systemu oraz określanie sieci quasi-relewantnej (ze względu na termy synonimiczne lub szersze i węższe) do danego termu. Operacja ta została nazwana przez konstruktora systemu – *zwiększaniem wiedzy adresatora*, i odpowiedzialna jest za nią funkcja *wiedzaAdresatora (v_term, v_baza)*, gdzie *v_term* oznacza poszukiwany term zaś *v_baza* określa nazwę dwuwymiarowej tabeli o polach typu znakowego (w formacie dBase III), do której będą zapisywane końcowe wyniki poszukiwań dla danego termu (patrz tabela 4.12).

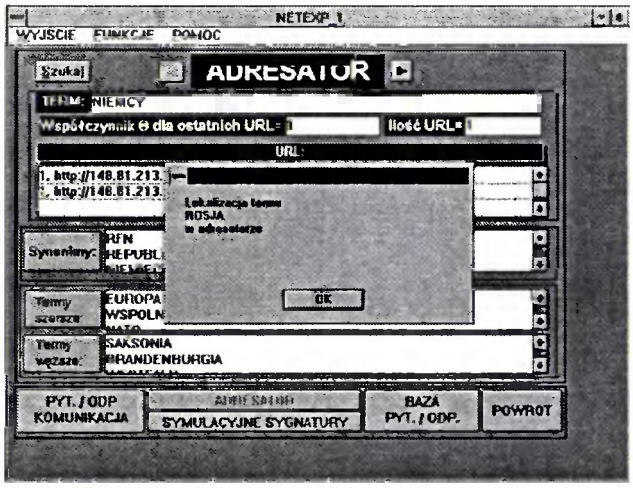
Tabela 4.12. Utworzenie pustej tablicy poszukiwanego termu

TERM_TEMP	ADRES_TEMP
	(...)

Najpierw funkcja sprawdza położenie termu w adresatorze w celu ustalenia *ID* strony[124], w przypadku nie znalezienia termu jest on dodawany i automatycznie ustalany jest numer strony (*PageRef*) dla nowo wprowadzonego termu.

[123] Za zwiększanie wiedzy adresatora odpowiedzialna jest funkcja *wiedzaAdresatora(v_term,v,baza)*, (por. materiały naukowe biblioteki Instytutu Informacji Naukowej i Studiów Bibliologicznych UW).

[124] Każdy obiekt w systemie ToolBook ma niepowtarzalne *ID*, czyli numer identyfikacyjny. Omawiana funkcja zwraca jako wartość czyli *ID* strony.



Rys. 4.23 Poszukiwanie termu w adresatorze

Istotną częścią funkcji jest omówiona wcześniej funkcja przeszukująca sygnatury *szukajTermAdrSyg ()* lub jej uogólnienie dla wszystkich sygnatur *szukajTermAllSyg (),* które zostały omówione wcześniej. Jeśli wynikiem działania funkcji *szukajTermAllSyg ()* jest fałsz, co oznacza, że nie zostały znalezione w sygnaturach żadne termy relewantne do danego termu, to uruchamiana jest funkcja *szukajSynAllSyg ()* poszukująca adresy dla synonimów szukanego termu. Jeśli jej wynik jest równy *FALSE,* co oznacza, że nie zostały znalezione żadne adresy, to uruchamiana jest funkcja *szukajBNAllSyg ()* poszukująca adresy dla termów węższych i szerszych.

Po zlokalizowaniu adresów dla termów węższych oraz szerszych (ew. synonimów) likwidowane są powtarzające się adresy w bazie, następnie obliczany jest współczynnik aproksymacji termu i zawartość tabeli dodawana jest do adresatora. Dodawane są jednak albo różne adresy, albo adresy o różnym współczynniku aproksymacji termu.

Należy zaznaczyć, że system tworzy dla każdego termu osobną tabelę. Istotnym elementem funkcjonowania NetExp jest zlokalizowanie danego termu w adresatorze i ustalenie parametrów miejsca, gdzie się on znajduje[125] (patrz rys. 4.23). W przypadku nie znalezienia termu jest on automatycznie zapisywany do adresatora.

Końcowym efektem działania funkcji *wiedzaAdresatora (v_term, v_baza),* jest zapisanie w utworzonej tabeli zbioru termów i adresów oraz podanie współczynnika dokładności danego termu. Obecnie zostaną omówione poszczególne etapy działania tej funkcji.

[125] Chodzi o tzw. zmienną *PageRef,* która identyfikuje stronę adresatora, na której znajduje się dany term.

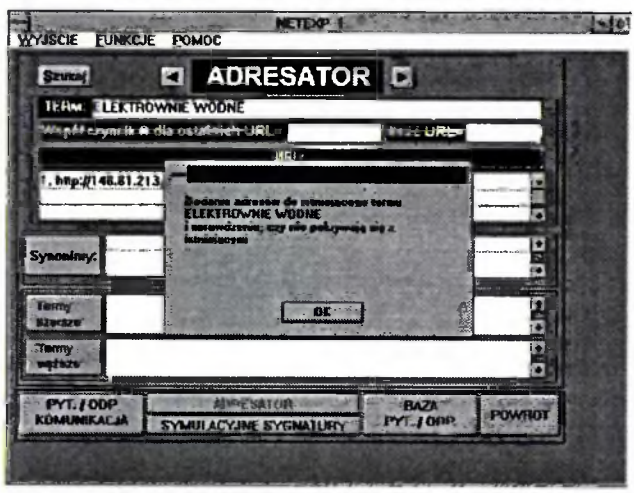
Uczenie się adresatora

Uczeniem się adresatora, omówionym w rozdziale 3, steruje funkcja *uczenieAdresatora (v_baza).* Po zlokalizowaniu wszystkich adresów relewantnych do danego termu i wszystkich termów o takich samych adresach, każdy znaleziony term porównywany jest ze zbiorem termów w adresatorze. Jeśli dany term istnieje w adresatorze, to dodawane są tylko adresy (patrz rys. 4.24). Jeśli term nie występuje, to dodawany jest i term i adres (patrz rys. 4.25).

Oczywiście system sprawdza, czy dodawane adresy nie pokrywają się z istniejącymi adresami przy danym termie. Po zakończeniu wyszukiwania termów i adresów wyświetlana jest informacja o ilości dodanych termów i adresów (patrz rys. 4.26). Ponieważ adresy te są w pełni adekwatne do wszystkich znalezionych termów, to współczynnik dokładności dla każdego termu wynosi 1.

Argumentami funkcji *uczenieAdresatora (v_baza)* jest baza danych utworzona po działaniu funkcji *szukajTermAllSyg ().* Tabela 4.13 przedstawia wynik działania funkcji dla termu *OCHRONA ŚRODOWISKA.*

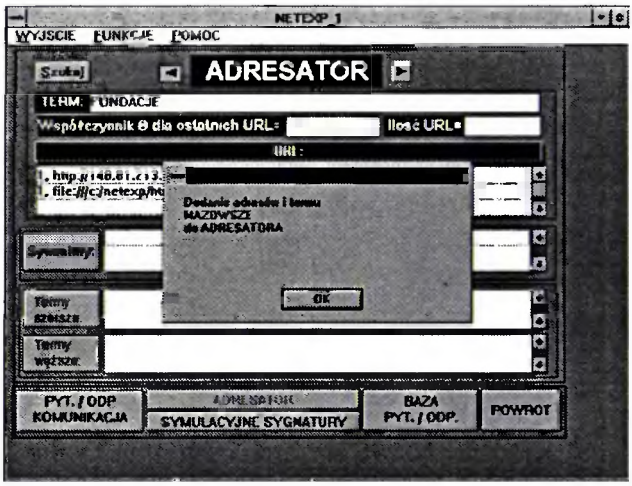
Z tabeli 4.13 dodawane są do adresatora: term oraz adresy. Najpierw lokalizowany jest dany term w adresatorze. Efektem działania operacji poszukiwania termu w adresatorze[126] jest zmienna *ID* strony w adresatorze. (zmienna *PageRef*) Jeśli zmienna *PageRef* przyjmie wartość *FALSE,* co oznacza, że term nie został znaleziony w adresatorze, wówczas poszukiwany term jest dodawany do adresatora. Funkcją odpowiedzialną za dodanie nowego termu do adresatora jest *dodajTermADR (),* zwraca ona *ID* strony, do której został wprowadzony nowy term. Następnie dodawany jest skojarzony z nowym termem adres[127].



Rys. 4.24 Dodanie adresów do istniejącego termu

[126] Poszukiwanie termów w adresatorze dokonuje się dzięki funkcji *szukajTermADR ().*

[127] Procedura: *put „1" & „ , " && v_adres_temp & CRLF after text of RecordFleld „URL" of PageRef,* (patrz materiały naukowe biblioteki Instytutu Informacji Naukowej i Studiów Bibliologicznych UW).



Rys. 4.25 Dodanie termu do adresatora

Rys. 4.26 Informacja o dodaniu termów i adresów

Jeśli term istnieje w adresatorze, system określa *ID* strony, następnie sprawdza, czy dany adres nie pokrywa się z istniejącym zbiorem adresów w adresatorze przy danym termie. W adresatorze adresy ulokowane są w polu danych (*Record-FIeld „URL"*). Pole *„URL"* jest polem wierszowym[128] z możliwością lokalizacji pojedynczych wierszy. System sprawdza każdy wiersz pola „URL" dla danej strony i porównuje z nowo wprowadzanym adresem. Wprowadzany jest tylko adres niewystępujący w polu „URL" lub o różnym współczynniku aproksymacji termu. W miarę wzbogacania się wiedzy będą mogły być wprowadzane takie same adresy o różnych współczynnikach, lecz nie mniejszych od wartości progowej ε.

[128] typ pola określony jest jako ang. *single selected line.*

Tablica 4.13. Tablica *„temp2.dbf"* dla termu *„OCHRONA ŚRODOWISKA"*

TERM_TEMP	ADRES_TEMP
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_2.htm
POLSKA	http://148.81.213.3/info_2.htm
MAZOWSZE	http://148.81.213.3/info_2.htm
ŚLĄSK	http://148.81.213.3/info_2.htm
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_4.htm
SAKSONIA	http://148.81.213.3/info_4.htm
NIEMCY	http://148.81.213.3/info_4.htm
OCHRONA ŚRODOWISKA	http://148.81.213.3/info_5.htm
ENERGIA SŁONECZNA	http://148.81.213.3/info_5.htm
FUNDACJE	http://148.81.213.3/info_5.htm
OCHRONA ŚRODOWISKA	http://148.81.213.5/info_6.htm
POLSKA	http://148.81.213.5/info_6.htm
NIEMCY	http://148.81.213.5/info_6.htm
MAZOWSZE	http://148.81.213.5/info_6.htm
ENERGIA SŁONECZNA	http://148.81.213.5/info_6.htm
ZANIECZYSZCZENIA	http://148.81.213.5/info_6.htm
ELEKTROWNIE WODNE	http://148.81.213.5/info_6.htm
POMORZE	http://148.81.213.5/info_6.htm

Opisana powyżej operacja powtarzana jest dla każdego termu i adresu z tablicy (temp1.dbf -dla termu pierwszego oraz temp2.dbf -dla termu drugiego).

4.4. WNIOSKI

W czasie eksperymentu zauważono, że proces uczenia się systemu NetExp jest dość szybki. Jedyną przeszkodą mogą być ograniczenia samej sieci Internet w procesie przeszukiwania sygnatur. Jednak dla znacznej ilości termów i dużej liczby sygnatur proces uczenie się systemu (jak również wzbogacanie wiedzy adresatora) może okazać się zbyt powolnym, dlatego może okazać się konieczne:

- napisanie systemu NetExp w dowolnym języku niskiego poziomu,
- zastosowanie komputerów o zwiększonych mocach obliczeniowych.

W następnym rozdziale zostanie omówiony element komunikacji systemu NetExp z siecią Internet.

5. WSPÓŁDZIAŁANIE SYSTEMU NETEXP Z SIECIĄ

5.0. WSTĘP

W poprzednim rozdziale omówiono system NetExp. Obecnie zostanie przedstawione współdziałanie tego systemu z siecią. Z uwagi na to, że żadna z istniejących sieci nie spełnia całkowicie warunków nakreślonych w zaproponowanym w tej pracy modelu, eksperyment współdziałania systemu NetExp przeprowadzono na drodze symulacyjnej. Przez symulację rozumiemy tutaj fakt, że sieć została zasymulowana na lokalnym komputerze. W ramach dalszych prac planuje się dostosowanie systemu NetExp do warunków jakie narzuca Internet. Wspomnijmy, że już obecnie opracowano pakiet funkcji programu, które odpowiedzialne są za kierowanie zlecenia do sieci Internet i przesyłanie odpowiednich zasobów informacyjnych do komputera użytkownika. Poniżej zostaną omówione elementy systemu NetExp związane z dostępem do sieci, z uwzględnieniem technologii i narzędzi stosowanych w Internecie.

5.1. POŁĄCZENIE Z SIECIĄ INTERNET

W celu zapewnienia poprawnego działania programu NetExp konieczne jest połączenie z siecią Internet. Proponuje się dwa rodzaje połączeń z siecią Internet:

- Przez sieć lokalną połączoną[129] z siecią Internet;
- Przez tzw. protokoły dwupunktowe[130] typu: SLIP[131] lub PPP[132].

Dla systemu NetExp konieczne jest wykorzystywanie mocy obliczeniowych komputera, na którym jest zainstalowany NetExp, dlatego zdecydowano się na przetestowanie systemu wykorzystując połączenie przez sieć lokalną[133]. Do tego

[129] Chodzi o połączenie danej sieci lokalnej przy pomocy urządzeń sieciowych (por. rozdz. 2).

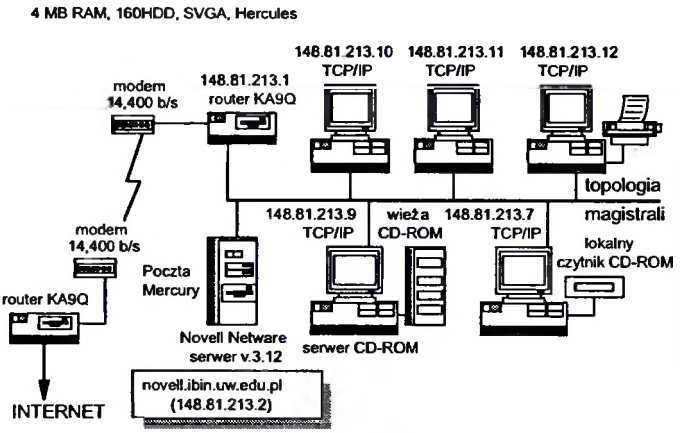
[130] Coraz powszechniej proponuje się dwa schematy służące przysyłaniu datagramów IP (podstawowych pakietów informacji w sieci Internet) przez publiczne łącza telefoniczne: SLIP (Serial Line Internet Protocol) oraz PPP (Point-to-Point-Protocol) [SHE95] s. 897

[131] Protokół PPP daje możliwość obsługi różnorodnych protokołów (nie tylko TCP/IP) [SHE95] s.897

[132] Protokół SLIP daje możliwość transmisji jedynie datagramów protokołów IP (Internet Protocol) [SHE95] s. 959.

[133] Możliwe było również korzystanie z sieci Internet przez protokoły dwupunktowe, jednak ze względów ekonomicznych (koszt połączenia telefonicznego) oraz ograniczeń wy-

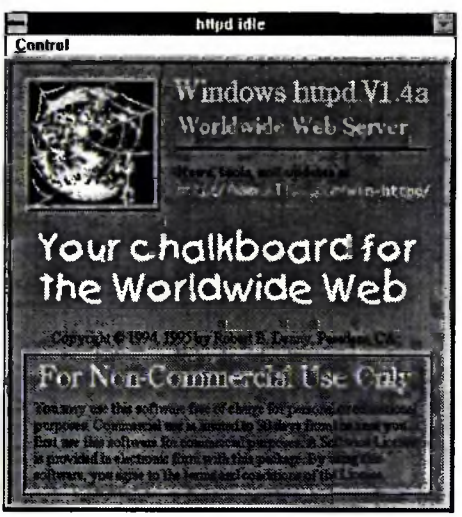
celu wykorzystano lokalną sieć komputerową połączoną do sieci Internet w Instytucie Bibliotekoznawstwa i Informacji Naukowej Uniwersytetu Warszawskiego (IBIN UW). Rysunek 5.1 przedstawia schemat laboratorium komputerowego IBIN UW oraz jego połączenie z siecią Internet.



Rys. 5.1. Schemat połączenia laboratorium komputerowego IBIN UW z siecią Internet

5.1.1. Serwery WWW

Do eksperymentu wykorzystano zainstalowane w laboratorium komputerowym IBIN UW serwery WWW pracujące w środowisku Windows – tzw. HTTPD (patrz rys. 5.2).



Rys. 5.2 Serwer WWW dla środowiska Windows

nikających z publicznych łącz telefonicznych zdecydowano się na połączenie przez sieć komputerową.

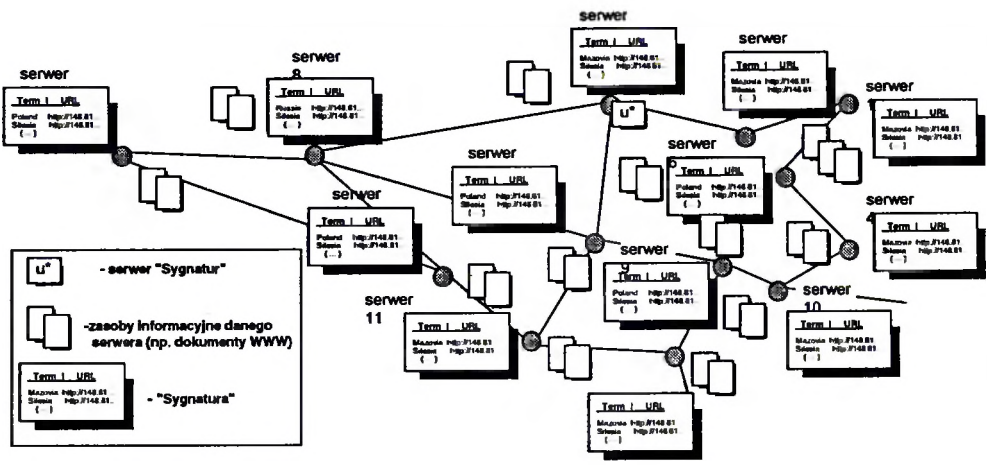
Przez serwer WWW rozumiane jest tu oprogramowanie umożliwiające jednoczesny dostęp wielu użytkowników sieci Internet do dokumentów zapisanych w języku HTML[134].

W przyszłości planowane jest utworzenie i zainstalowanie sygnatur na komputerach bazowych sieci Internet – wspomnianych serwerach WWW w laboratorium IBIN UW. Rysunek poniżej przedstawia serwer WWW dla środowiska Windows, który został wykorzystany w eksperymencie.

5.2. SYGNATURY

Jak zaznaczono w rozdziale 4 system NetExp nie dysponuje jeszcze realnymi sygnaturami, dlatego wprowadzono tzw. symulacyjne sygnatury. Problem sygnatur pozostaje jednak ciągle aktualny i ich wprowadzenie jest nieuniknionym etapem rozwoju systemu NetExp. Przypomnijmy, że sygnatury wraz z tzw. serwerem sygnatur stanowią serwer systemu NetExp[135].

Według definicji 3.21 przyjęto, że sygnatura jest sumą logiczną zbiorów wszystkich termów indeksujących wszystkie zasoby danego komputera bazowego. Sygnatury odgrywają istotną rolę w systemie NetExp, od nich bowiem zależy wynik wyszukiwania i działanie całego systemu. Rysunek 5.3 przedstawia schemat postulowanych sygnatur w środowisku sieciowym.



Rys. 5.3 Sygnatury w systemie NetExp

[134] W przypadku serwera HTTPD maksymalna liczba użytkowników mogących uzyskać dostęp do zasobów jednego serwera WWW wynosi 16.

[135] Por. podrozdz. 2.5 Sieć Internet, Architektura klient-serwer.

5.2.1. Automatyczne tworzenie sygnatur

Rzeczywiste sygnatury są na razie propozycją. Tutaj wypada tylko sformułować postulat do osób odpowiedzialnych za rozwój sieci Internet o stworzenie łatwo dostępnych i przyjaznych dla użytkownika narzędzi do tworzenia sygnatur. Jednym z możliwych rozwiązań byłoby utworzenie automatycznego systemu pozwalającego na przeglądanie zasobów danego serwera i jednoczesne tworzenie sygnatur na podstawie wystąpień słów kluczowych w danych dokumentach.

Ze względu na dość dużą popularność zasobów informacyjnych WWW mechanizm obsługi sygnatur powinien być zastosowany najpierw do zasobów informacyjnych WWW. Automatyczne tworzenie sygnatur polegałoby na przeszukiwaniu wszystkich stron WWW i wybieraniu słów kluczowych. Następnie słowa kluczowe i adresy URL danych stron WWW byłyby umieszczane w sygnaturach. Ze względu na dość szybkie zmiany zachodzące w sieci, mechanizm ten powinien być automatyczny i posiadać czasowy wyłącznik. Powinien być uruchamiany np.: raz na tydzień albo raz na dzień, lub w porach najmniejszego „natężenia ruchu" w sieci.

Mechanizm aktualizacji i tworzenia indeksów zbiorów powinien być dwustopniowy:

1) na podstawie informacji pochodzącej z systemu operacyjnego danego komputera system powinien sprawdzać, czy jakieś strony WWW uległy zmianie od czasu ostatniego tworzenia/aktualizacji sygnatury lub/i czy stworzono nowe strony WWW,
2) jeśli stwierdzono zmiany, mechanizm tworzenia sygnatury przeglądałby tylko te zbiory, które uległy zmianie lub/i te strony, które zostały ostatnio stworzone i zapisywałby nowe indeksy w sygnaturach.

Zarysowany tutaj postulowany mechanizm automatycznego tworzenia sygnatury wymaga odpowiedniego współdziałania z danym serwerem WWW oraz z systemem operacyjnym, na bazie którego zainstalowany jest dany serwer WWW.

5.2.2. Serwer sygnatur

Wśród sygnatur istotną rolę odgrywa tzw. *serwer sygnatur.* Od informacji, które przechowuje w swojej bazie zależy, które sygnatury będzie przeszukiwać system NetExp. Na razie rolę serwera sygnatur w systemie NetExp pełni zbiór tekstowy *hosty.txt,* do którego użytkownik może wpisać nazwy symulacyjnych sygnatur, które mają być przeszukiwane. Na obecnym etapie rozwoju systemu NetExp serwer sygnatur może być utożsamiany do pewnego stopnia z serwerem dostępu[136]. Należy jednak pamiętać, że serwer dostępu jest pierwszym komputerem w sieci, z którym użytkownik ma bezpośredni kontakt.

Serwer sygnatur powinien być komputerem o dużych mocach obliczeniowych i możliwości obsługi jak największej liczby użytkowników systemu NetExp w sie-

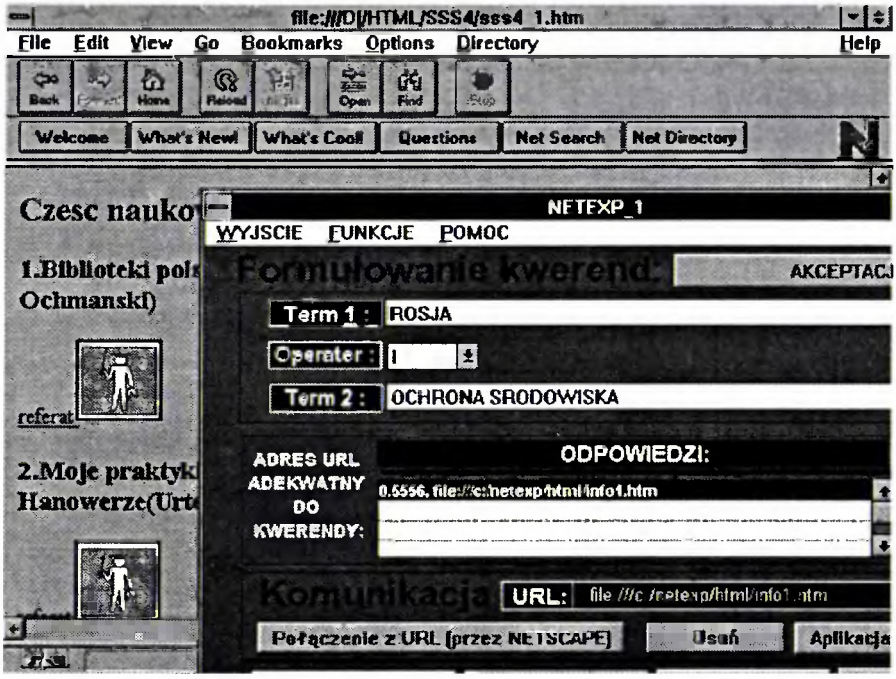
[136] Por. rozdz. 3 i rozdz.4. oraz def. 3.2.

ci. Ponadto powinien on posiadać dostęp do jak najszybszych *„arterii"* komunikacyjnych sieci Internet. Serwer sygnatur powinien oprócz tego automatycznie zbierać informacje na temat nowo powstałych sygnatur. Oznacza to, że same sygnatury powinny w momencie swego zainstalowania i przetestowania wysyłać informacje do wszystkich serwerów sygnatur w celu ich zarejestrowania.

Należy jednak pamiętać, że zmiany w sieci dotyczą nie tylko powstawania nowych zasobów i ich aktualizacji, ale także możliwości komunikacyjnych samych sieci komputerowych[137].

5.3. KOMUNIKACJA Z SIECIĄ W SYSTEMIE NETEXP

Komunikacja z siecią dotyczy wybrania adresów URL o najwyższym współczynniku odpowiedzi i automatycznego połączenia się z nimi przez Netscape. Połączenie jest możliwe dzięki wykorzystaniu nieznacznie zmodyfikowanych funkcji autorstwa Paolo Tossoliniego z Uniwersytetu w Trieście (tzw. *MMWWWPC*). Rysunek 5.4 przedstawia moduł komunikacji w systemie NetExp. Komunikacja jest możliwa dzięki klientowi WWW – Netscape, który pełni rolę serwera dla systemu NetExp.



Rys. 5.4 Moduł komunikacji z wybranymi zasobami w NetExp

[137] Chodzi o przepustowość sieci komputerowych i czasową niedostępność niektórych komputerów w sieci Internet (por. def. 3.7 i def. 3.8).

Do najważniejszych funkcji należy funkcja *getURL* (*nazwa URL*). Wynikiem jej działania jest połączenie się ze wskazanym adresem URL przez Netscape[138].

Chociaż Netscape i Mosaic służą przeważnie do łączenia się z serwerami WWW, to można je wykorzystywać do przeglądania innych zasobów informacyjnych np. Gopher, WAIS czy nawet do przesyłania zbiorów z odległego komputera bazowego na stację roboczą użytkownika[139]. Dzięki URL można również zlokalizować zbiory znajdujące się na dysku lokalnym[140], co okazało się pomocne w fazie testowania systemu NetExp.

Po uruchomieniu tej funkcji aktywną aplikacją staje się Netscape i następuje automatyczna próba połączenia ze wskazanym adresem URL. Komunikacja z wybranymi adresami URL jest możliwa z następujących modułów:

- pytania/opowiedzi/komunikacja,
- baza pytania/odpowiedzi,
- adresator.

Po wybraniu odpowiednich adresów URL użytkownik wskazuje je przyciskiem myszy, co powoduje wywołanie okna dialogowego (patrz rys. 5.6).

W systemie nie zostały wykorzystane[141] wszystkie opisane niżej funkcje, jednak w dalszych fazach testowania systemu, w zależności od potrzeb użytkowników programu, będą one stopniowo wprowadzane.

5.3.1. Projekt MM-WWW-PC

Projekt MM-WWW-PC jest próbą wykorzystania możliwości serwerów WWW oraz mocy obliczeniowych współczesnych komputerów osobistych[142].

MM-WWW-PC składa się z szeregu funkcji pomocnych w tworzeniu aplikacji systemu ToolBook dostępnych w sieci Internet. Dzięki projektowi MMWWW-PC aplikacje systemu ToolBook wydają się szczególnie użyteczne w przypadku odległego szkolenia czy prezentacji informacji hipermedialnych w sieci Internet. Rysunek 5.5 przedstawia aplikacje z podstawowymi funkcjami MM-WWW-PC. W pewnym stopniu system ten jest udoskonaleniem możliwości typowych klientów (przeglądarek) WWW.

---

[138] Definicję całej funkcji znajdzie czytelnik w materiałach naukowych bibioteki Instytutu Informacji Naukowej i Studiów Bibliologicznych UW.

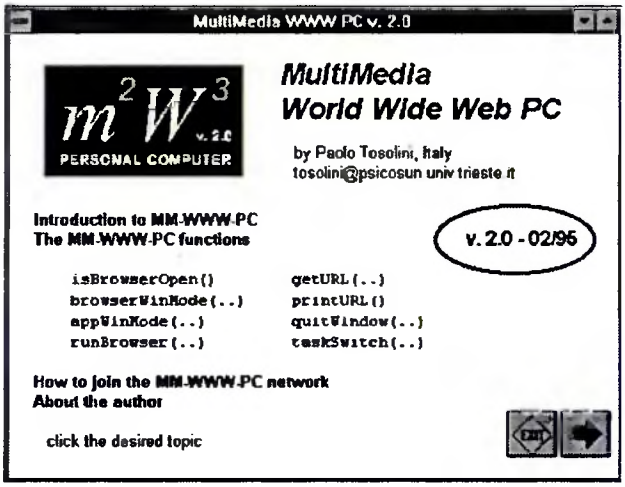
[139] Por. podrozdz. 2.5 oraz podrozdz. 3.2.

[140] Np.: adres w konwencji URL dla zbioru – *c:/netexp/htm/info.htm* wyglądałby następująco: *file:///c:/netexp/html/info1.htm.*

[141] Obecnie funkcje te są niedostępne z poziomu użytkownika, jednak istnieje możliwość ich prostego zaimplementowania przez projektanta systemu, bowiem zostały one już wcześniej zdefiniowane w systemie NetExp.

[142] Chodzi o wykorzystanie przyjaznych (dla projektanta) systemów do tworzenia aplikacji środowiska Windows.

MM-WWW-PC wymaga zmian konfiguracyjnych danej przeglądarki WWW[143], czyli NETSCAPE lub MOSAIC oraz niewielkich zmian konfiguracyjnych w środowisku Windows i systemie MS-DOS. Projekt MM-WWW-PC obejmuje zainstalowanie aplikacji ToolBook wykorzystujących funkcje MM-WWW-PC w serwerach WWW, przez co konieczne stają się dodatkowe zmiany na serwerach WWW.



Rys. 5.5 Aplikacja z funkcjami projektu MM-WWW-PC.

Przez daną przeglądarkę WWW (NETSCAPE, MOSAIC) użytkownik łączy się z odpowiednią stroną WWW i przez wybranie wskazanego zbioru o rozszerzeniu *tbk* może przeglądać daną aplikację ToolBook'a. Wersja 2 MM-WWW-PC (której funkcje wykorzystano w systemie NetExp) pozwala na współpracę ze wszystkimi wersjami programu Netscape oraz z 32-bitową wersją programu Mosaic. Funkcje MM-WWW-PC przygotowane były dla multimedialnych aplikacji systemu ToolBook (tzw. Multimedia ToolBook v.3). Dzięki programowi konwersji mtb2tb.tbk dostępnemu przez anonymous ftp z komputera bazowego asymetrix.com dokonano konwersji na aplikację systemu ToolBook, przez co możliwe było zastosowanie opisanych w tym rozdziale funkcji.

5.3.2. Rola przeglądarek WWW w NetExp

Należy zaznaczyć, że istotną rolę w procesie komunikacji w systemie NetExp odgrywają typowe przeglądarki WWW. Do takich zalicza się w tej pracy:

- NETSCAPE (od wersji 1.0),
- MOSAIC (dla wersji 32-bitowych)

[143] Przez przeglądarkę WWW będziemy rozumieli klienta systemu WWW (czasami nazywanego czytnikiem WWW, por. [MAR95]).

W fazie opracowywania modułu komunikacji istniały sugestie, aby zbudować niezależny moduł przesyłania i przeglądania dokumentów WWW, jednak zdecydowano się na wykorzystanie klasycznych przeglądarek WWW głównie ze względu na ich popularność. Prawdopodobnie w przyszłości klient systemu NetExp będzie posiadał własną przeglądarkę WWW i moduł komunikacji lub też wybrane elementy systemu NetExp zostaną wprowadzone do kolejnej wersji typowych przeglądarek NETSCAPE lub MOSAIC.

W celu uruchomienia danego klienta WWW wykorzystuje się charakterystyczną dla środowiska Windows możliwość współpracy wielu programów jednocześnie (wielozadaniowość) oraz architekturę klient-serwer w środowisku Windows. System NetExp jest klientem, który zleca innej aplikacji środowiska Windows wykonanie określonego działania. W przypadku komunikacji z serwerem WWW serwerem jest uruchomiony program Mosaic lub Netscape.

Funkcja *isBrowserOpen()*

Przez wywołanie funkcji: *IsMosaicOpen ()* oraz *IsNetscapeOpen ()* system NetExp sprawdza, która przeglądarka WWW jest aktywna. Funkcja *IsBrowseOpen ()* w przypadku działania programu Netscape zwraca ciąg znaków – *„NETSCAPE"*, a w przypadku programu MOSAIC zwraca ciąg znaków – *„MOSAIC"*, jeśli żadna z wyżej wymienionych aplikacji nie jest otwarta, funkcja zwraca wartość *„FALSE"*. Uruchamianie danej przeglądarki WWW odbywa się po wydaniu polecenia runBrowser.

browserWinMode (),

Funkcja *browserWinMode()* zmienia pozycję przeglądarki i jej stan. Funkcja ta sprawdza, czy jest otwarta przeglądarka WWW[144]. Jeśli jest już uruchomiony program Netscape lub Mosaic, to następuje odwołanie do następujących funkcji (dynamicznych bibliotek połączeń – DLL) środowiska Windows:

- zarządzania zbiorami, funkcji wejścia/wyjścia (linkDLL KERNEL),
- interfejsu użytkownika w środowisku Windows (linkDLL USER).

Dzięki podstawowym funkcjom środowiska Windows funkcja *browserWinMode ()* kontroluje położenie okna przeglądarki WWW. Funkcja ta jest funkcją od trzech argumentów:

- *APPhWnd* (element stały),
- *winState* – określa rozmiar okna (minimalny - *min,* maksymalny - *max,* normalny - *normal*),
- *WinPos* – określa pozycje okna (w tle- *back,* zawsze na aktywna *ontop* , aktywna- *front*).

[144] Por. funkcja *isBrowserOpen ()*.

appWinMode()

Funkcja *appWinMode()* jest analogiczna do funkcji *browserWinMode()* i kontroluje pozycję aplikacji ToolBook'a[145] i jej stan. Podobnie jak w funkcji *appWinMode()* następuje odwołanie do funkcji (dynamicznej biblioteki połączeń-DLL) środowiska Windows:

- zarządzania zbiorami, funkcji wejścia/wyjścia (*linkDLL KERNEL*),
- interfejsu użytkownika w środowisku Windows (*linkDLL USER*).

Dzięki podstawowym funkcjom środowiska Windows funkcja *appWinMode()* kontroluje położenie okna przeglądarki WWW. Funkcja ta jest funkcją od trzech argumentów:

- *sysWindowHandle* (element stały),
- *winState* – określa rozmiar okna (minimalny - *min,* maksymalny - *max,* normalny - *normal*),
- *WinPos* – może przybierać wartości *TRUE* lub *FALSE* i określa pozycje okna (*TRUE* – aplikacja zawsze na aktywna *ontop*).

5.3.3. Łączenie się z wybranym adresem URL

Połączenie z wybranym adresem URL dokonuje się dzięki funkcji *runBrowser ()* (patrz Rys. 5.6).

runBrowser()

Funkcja ta powoduje uruchomienie przeglądarki WWW wraz z podaniem adresu *URL* (jeśli jej nazwa jest w zmiennej *PATH*) . Zwraca wartość *TRUE,* jeśli program *Mosaic* lub *Netscape* został znaleziony na dysku. Argumentami funkcji są:

- podany w cudzysłowach adres URL,
- *mode* może przyjmować wartość FALSE lub TRUE, gdy przeglądarka ma być uruchomiona z minimalną ilością funkcji (ang. *kiosk mode*).

getURL()

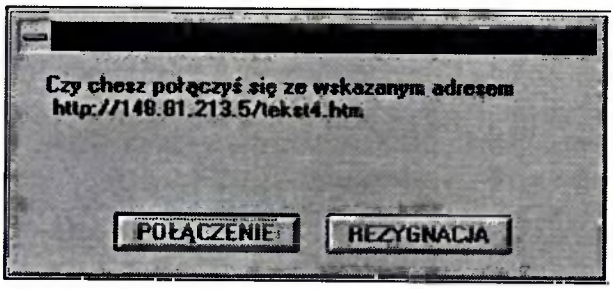
Funkcja ta jest podstawową funkcją wykorzystaną w procesie komunikacji z siecią Internet w modułach: pytania/odpowiedzi/komunikacji, baza/pytań/odpowiedzi i adresator. Funkcja ta odwołuje się do biblioteki funkcji *tb30win.dll* i do pozostałych funkcji MM-WWW-PC (*runBrowser(), BrowserWinMode ()*.

Wydruk stron WWW – printURL()

Dzięki funkcji *printURL ()* istnieje możliwość wydruku stron WWW przez moduł wydruku w danej przeglądarce WWW. Funkcja *printURL ()* wydrukowuje

[145] Czyli systemu NetExp.

aktualnie wyświetlaną przez daną przeglądarkę WWW stronę. W przypadku, gdy żadna z przeglądarek WWW nie działa, funkcja ta zwraca wartość *FALSE*.



Rys. 5.6 Komunikacja w modułach „Adresator" i baza/pytania/odpowiedzi

5.3.4. Praca z pozostałymi aplikacjami środowiska Windows

taskSwitch()

Funkcja ta pozwala na przechodzenie między działającymi aplikacjami środowiska Windows. Argumentem funkcji jest jeden parametr – *mode,* który przyjmuje wartość *TRUE* lub *FALSE*.

TRUE pozwala na przełączanie między poszczególnymi aplikacjami środowiska Windows *FALSE* bądź uniemożliwia przełączanie między poszczególnymi aplikacjami środowiska Windows.

quitWindow()

Funkcja ta służy do zakończenia pracy z działającą przeglądarką WWW lub z działającą aplikacją ToolBook´a (NetExp). Argumentami funkcji są:

browser przyjmuje wartość *TRUE,* jeśli ma być zakończona praca z daną przeglądarką WWW,

otherApps przyjmuje wartość *TRUE,* jeśli ma być zakończona praca z wszystkimi aplikacjami systemu ToolBook,

currentApp przyjmuje wartość *TRUE,* jeśli ma być zakończona praca z bieżącą aplikacją systemu ToolBook.

6. ZAKOŃCZENIE

W pracy wykazano, że jest możliwe stworzenie samouczącego się modelu pozwalającego na nawigację w niejednorodnych sieciach informacyjnych. Stworzono model matematyczny i komputerowy. Ich przydatność została potwierdzona eksperymentalnie. Dalsze prace nad modelem i systemem NetExp będą dotyczyły przede wszystkim: quasi-tezaurusa; sygnatur; procesu uczenia się; dostępu do innych sieci.

Quasi-tezaurus czyli część Adresatora[146] może być postrzegany bardziej jako narzędzie wspomagające twórcę tezaurusa niż kompletny tezaurus. Utworzony przykładowy quasi-tezaurus nie posiada bowiem części systematycznej, co może utrudniać proces wyszukiwania i gromadzenia dokumentów. Zdaniem specjalistów tezaurus powinien posiadać zarówno część alfabetyczną jak i część systematyczną[147]. Autor zetknął się też z zarzutami, że przedstawiony tezaurus ma charakter „prywatny", czyli nie odwołuje się do powszechnie uznanych standardów[148]. W odpowiedzi na ten ostatni zarzut, zostanie w przyszłości wprowadzony i przetestowany jeden z utworzonych przez specjalistów tezaurusów, oprócz tego quasi-tezaurus zostanie wzbogacony o część systematyczną połączoną z częścią alfabetyczną. W celu efektywniejszego działania systemu oraz pragnąc uniknąć zalewu adresatora niepotrzebnymi termami, w czasie procesu uczenia proponuje się wprowadzenie ograniczeń odnośnie do wprowadzanych termów. I tak do adresatora będą wprowadzane tylko te termy, które dotyczą konkretnej dyscypliny[149]. Wydaje się również koniecznym zwiększenie liczby relacji[150] m.in. o relację kojarzeniową (ang. *related term*).

W przyszłości prowadzone będą też prace nad rozszerzeniem procesu uczenia się systemu dla relacji między termami w adresatorze. W chwili obecnej proces uczenia się systemu zarówno w modelu matematycznym jak i w komputerowym ograniczony jest wyłącznie do wprowadzania termów bez możliwości łączenia termów relacjami[151].

---

[146] Por. rozdz.3 i 4.

[147] Chodzi o termy ułożone w porządku logicznym.

[148] W tym przypadku chodzi o odwołanie się do istniejących tezaurusów, bowiem zastosowanie „prywatnego" tezaurusa przeczy samej istocie tezaurusa.

[149] Np.: prawa, ekonomii, ochrony środowiska itp.

[150] Obecnie quasi-tezaurus posiada tylko trzy relacje: synonimii, terminy węższe, terminy szersze.

[151] Np.: synonimii, relacji węższy, szerszy.

W przedstawionym modelu istotną rolę odgrywają sygnatury[152]. Ich istnienie jest warunkiem dalszego rozwoju systemu NetExp, dlatego dalsze badania powinny skupić się wokół utworzenia sygnatur, ewentualnie wokół tworzenia odpowiednich interfejsów do już istniejących zaindeksowanych zasobów w sieci Internet.

W omówionym w rozdziale 5 modelu istotną rolę odgrywa tzw. serwer sygnatur. Mimo że współczesne komputery stają się maszynami o coraz większych mocach obliczeniowych, to w przypadku procesu uczenia się adresatora mogą okazać się one maszynami o zbyt ograniczonych parametrach, dlatego proponuje się przenieść cały proces uczenia się na odpowiedni serwer w sieci Internet. Zakładając, że nie jest możliwe stworzenie *uniwersalnego tezaurusa,* postuluje się też wyodrębnienie kilku zaproponowanych przez specjalistów tezaurusów i ograniczenie procesu uczenia się tylko do tych terminów.

Kolejnym istotnym elementem w pracy nad rozwojem systemu NetExp jest zwiększenie jego szybkości. Nie jest wykluczone, że zastosowanie języków programowania niskiego poziomu do utworzenia opisanego w rozdziale 3 modelu pozwoliłoby na stworzenie systemu znacznie szybszego niż obecny NetExp. Ponadto konieczne wydaje się zastosowanie lepszej platformy sprzętowej niż omawiana w rozdziale 4 i 5. Z uwagi na to, że opisany w niniejszej pracy model jest zastosowaniem architektury klient-serwer, dlatego prace nad zwiększeniem efektywności systemu NetExp powinny być prowadzone równolegle z tworzeniem i testowaniem sygnatur[153].

Obecnie system NetExp dotyczy wyłącznie sieci Internet a w niej sieci serwerów WWW. W przyszłości rozważone zostaną próby implementacji systemu do innych sieci informacyjnych i komputerowych[154] oraz do odmiennych systemów operacyjnych[155].

Praca nad przedłożoną w tej rozprawie koncepcją zachęciła do refleksji na temat zmiany podejścia i rozumienia podstawowych zagadnień w odniesieniu do systemów informacyjnych. Zauważmy na przykład, że pojęcie „relewantności", które w tradycyjnych systemach informacyjnych jest dobrze zdefiniowane[156] i ma swoje miary, w przypadku sieci traci swoje znaczenie[157]. Rozległość sieci, ogrom jej zasobów i spowodowana tym niemożliwość zlokalizowania wszystkich dokumentów, które odpowiadają zadanym pytaniom uniemożliwia obliczenie relewantności w klasycznym sensie. Przykładów tego typu można podać więcej. Nie jest to jednak przedmiotem tej pracy. Wydaje się jednak, że w dalszych działaniach praktycznych, które opisano wyżej w tym rozdziale warto pamiętać o „zmieniającym się paradygmacie" współczesnych systemów informacyjnych.

---

[152] Por. rozdz.: 3,4,5

[153] Por. uwagi wcześniejsze uwagi na temat sygnatur.

[154] Chodzi o sieci publiczne np.: X-25 , oraz serwisy informacyjne np.: Dialog-Datastar.

[155] Np. komputery pracujące z systemem operacyjnym UNIX.

[156] Por. [SAL77], [SAL71].

[157] Na sprawę tę zwróciła uwagę prof. A. Sitarska podkreślając potrzebę nowego spojrzenia na sposób definiowania relewantności.

Streszczenie

Sprawne pozyskiwanie informacji odgrywa coraz większą rolę w różnorakich procesach zachodzących we współczesnych, rozwiniętych społeczeństwach. Źródła informacji są zazwyczaj rozproszone i niejednorodne, ich liczba stale i szybko rośnie. Skomputeryzowane, rozległe sieci informacyjne stają się kluczowym elementem w szerokiej gamie narzędzi i środków używanych do bezzwłocznego docierania do zasobów informacyjnych, prowadzenia w nich wyszukiwania i sprowadzania relewantnych danych bezpośrednio do miejsca pracy użytkownika. Przykładem takiej sieci jest Internet, który w coraz większym stopniu jest wykorzystywany w pracy humanistów, techników, przedsiębiorców, urzędników i wszystkich tych, dla których dostęp do informacji jest warunkiem koniecznym należytego wykonywania obowiązków.

Ogromna ilość informacji oraz bardzo duża liczba baz danych sprawiają, że efektywne poruszanie się w sieci (nawigowanie) i odnajdywanie relewantnych dokumentów stanowią jedne z najważniejszych problemów pojawiających się w kontekście korzystania z sieci i jej zasobów. Każde usprawnienie w tym względzie jest szczególnie cenne. Niniejsza praca jest próbą z tego zakresu: zawiera ona model sieci informacyjnej oraz samouczącego się mechanizmu do wspomagania procesów nawigowania i wyszukiwania informacji w sieci. Zaproponowany mechanizm opiera się na *quasi*-tezaurusie, którego elementami są termy z przypisanymi do nich adresami URL (ang. *Uniform Resource Locator*) lokalizującymi źródła informacji (bazy danych, dokumenty), gdzie termy te występują. Każde zlecenie, przed „wysłaniem" go do sieci, jest „przetwarzane" przez quasi-tezaurus w ten sposób, że występujące w nim termy odnajdywane są w *quasi*-tezaurusie i zaopatrywane w odpowiadające im adresy URL. Opracowano specjalną procedurę w przypadku, gdy lista adresów URL związana z termem *quasi*-tezaurusa jest pusta. Po uzyskaniu zbioru dokumentów stanowiących odpowiedź, wszystkie termy indeksujące te dokumenty i jednocześnie nie należące do pytania, zostają użyte do aktualizacji quasi-tezaurusa. W tym właśnie sensie zachodzi proces uczenia się *quasi*-tezaurusa na podstawie informacji pochodzących z zasobów informacyjnych sieci.

Opracowany model matematyczny został zrealizowany w postaci samouczącego się systemu komputerowego, który zbudowano za pomocą obiektowego, hipertekstowego języka programowania OpenScript (systemu ToolBook v.3.0) w architekturze klient-serwer. W laboratorium komputerowym Instytutu Informacji Naukowej i Studiów Bibliologicznych Uniwersytetu Warszawskiego (dawniej Instytut Bibliotekoznawstwa i Informacji Naukowej Uniwersytetu Warszawskiego) przeprowadzono eksperyment wykazujący poprawność i przydatność utworzonego modelu komputerowego.

Wychodząc z założenia, że rozwój skomputeryzowanych sieci informacyjnych i dostępnych w nich narzędzi wyszukiwawczych uzależniony jest zarówno od rozwoju środków technicznych, jak i żądań i potrzeb użytkowników, w pracy omówiono również zasady działania złożonych sieci informacyjnych oraz rolę inteligentnych systemów komunikacji człowiek-komputer w procesie wyszukiwania informacji.

Summary

The information resources available through the Internet are immense. The total volume of the files accessible via the Internet is counted in thousands of Gigabytes. Therefore, the main problem connected to the Internet is the flood of information. The most obvious is the difficulty of simply finding items in the vast seas of available material. Therefore, the issue of identifying the relevant resources and accessing them is of crucial importance, in particular from casual users' perspective. As a rule users' specific knowledge about the network itself, and the distribution and contents of the information sources is limited. This paper addresses this issue by proposing a method helping the users to establish „good" queries and submitting them to the information resources (residing on the network) which are likely to contain the relevant documents. The core idea of the proposed approach is to create a self-learning mechanism supporting the queries' establishment and forwarding them to the right places within the network. The mechanism, called *ADDRESSER,* has been conceived as a simple quasi thesaurus which is composed of terms, URL addresses related to the terms and relations linking the terms. A prototype for testing and evaluating the idea was implemented. The experiments have proved the idea is viable, sound and workable.

LITERATURA

W spisie literatury zastosowano opisy skrócone źródeł wystarczające do zidentyfikowania poszczególnych pozycji.

[ABB94] Abbott T.: Internet World's on Internet 94.An International Guide to Electronic Journals, Newsletters, Texts, Discussion Lists, and Other Resources on the Internet. Westport 1994.

[AND93] Anderson C.: The Rocky Road to a Data Highway. Science May 21 1993.

[ANH94] Angell D., Heslop B.: The Elements of E-mail Style. New York 1994.

[ARM90] Arms C. R.: A New Information Infrastructure. Online September 1991 vol. 14 nr 5 s. 15.

[BEC73] Becker H.: Functional Analisys of Information Networks. Amsterdam [i in.]1973.

[BGS95] Brown C., Gasser L., O'Leary D. E., Sangster A.: AI on the WWW, Supply and Demand Agents. „IEEE Expert" August 1995 s. 50-55.

[BIE87] Bieńkowska B.: Zarys dziejów książki. Warszawa 1987.

[BIE89] Bieńkowska B.: Metody bibliologiczne w badaniach dziejów nauki. Kwartalnik Historii Nauki i Techniki 1989 nr 2 s.331-342.

[BOR77] Samuelson K., Borko H., Amey G.X.: Information Systems and Networks. Amsterdam, New York, Oxford 1977.

[BOS84] Słupecki J., Borkowski L.: Elementy logiki matematycznej i teorii mnogości. Warszawa 1984.

[BRA94] Braun E.: The Internet Directory. New York 1994.

[BRO94] Browne S.: The Internet via Mosaic and World-Wide Web. Emeryville 1994.

[CIE95] Using the Internet for Issue Oriented Information Retrieval, The Environmental and Sustainable Development. A training workshop sponsored by CIESIN and the Sustainable Develpment Networking Programme. Warsaw 1995.

[CHD84] Charniak E., McDermott D.: Introduction to Artificial Intelligence,. Menlo Park [i in.] 1984.

[COG95] Corrigan P., Guy A.: Budowa lokalnych sieci komputerowych Novell Netware. Tł. [ang.] Warszawa 1993.

[COM93] Comer D. E.: Internet working with TCP/IP vol. 1-3. Eglewood Clifs, N. J. 1993.

[CRO95] Croft B.: NSF Center for intelligent Inforamation Retrieval. Communication of the ACM 1995 nr 4 s.42-43

[DRZ91] Drzewiecki M.: Biblioteka we współczesnej szkole. Warszawa 1991

[DRZ90] Drzewiecki M.: Biblioteki szkolne i pedagogiczne w Polsce, rola w procesie dydaktycznym i miejsce w krajowej sieci biblioteczno-informacyjnej. Warszawa 1990

[EAR93a] Listserv User Guide. EARN Association 1993 (bez miejsca wydania)

[EAR93b] Guide to Network Resource Tools. EARN Association 1993 (bez miejsca wydania)

[ENG93] Engle M.: Internet Connections. A librarian guide to dial-up access and use. Chicago 1993.

[EST94] Estrada S.: Connecting to the Internet. Sebastopol 1994.

[ETW95] Etzioni O., Weld D. S: Intelligent Agents on the Internet. Fact, Fiction, and Forecast. „IEEE Expert" August 1995 s.44-49.

[FAKL95] Fox E. [et al]. Digital Library. Communication of the ACM 1995 nr 4 s. 24-28

[GEP70] Galler B. A., Perlis A. J.: A view of programming languages. Menlo Park 1970.

[GIB93] Gibbs M.: Sieci komputerowe, biblia użytkownika. Warszawa 1994.

[GIL94] Gilster, P.I: Finding it on the Internet. The Essential Guide to Archie, Veronica, Gopher, WAIS, WWW (Including Mosaic), and Other Search and Browsing Tools. New York 1994.

[GLI93] Gliński W.: Lokalna sieć komputerowa w dydaktyce. Praktyka i Teoria Informacji Naukowej i Technicznej 1993 nr 3-4 s.39-41.

[GLI94] Gliński W.: Integracja laboratorium komputerowego z siecią Internet. Praktyka i Teoria Informacji Naukowej i Technicznej 1994 nr 4 s.31-38.

[GLI95a] Gliński W.: Programy wspomagające działanie WWW. Praktyka i Teoria Informacji Naukowej i Technicznej 1995 nr 2 s.40-44.

[GLI95b] Gliński W.: Podejście obiektowe w projektowaniu systemów hipermedialnych. Informacja Profesjonalna. 1995 nr 3 s.34-38.

[GLI95c] Gliński W.: Laboratorium komputerowe w dydaktyce Instytutu Bibliotekoznawstwa i Informacji Naukowej UW. W: Bibliotekoznawstwo i Informacja Naukowa. Kształcenie w perspektywie nowego stulecia. Red. E. B. Zybert. Warszawa 1995 s.105-126.

[HAS94] Hahn H., Stout R.: The Internet Yellow Pages. Berkeley, Osborne 1994.

[HEL93] Held G.: Internetworking LANs and WANs. Concepts, techniques and methods. Chichester 1993.

[KOC93] Kochmer J.: Internet passport. North WestNet's guide to our world online. Bellevue 1993.

[KBA91] Khosafian S., Baker B. A., Abnous R., Shepherd K.: Intelligent offices. Object-Oriented Multimedia-Media Information Management in Client-/Server Architectures. New York 1991.

[KRO93] Krol E.: The whole Internet user's guide & catalog. Sebastopol. CA. 1993.

[KRZ93] Krzanowski W.: Aktualne możliwości korzystania z NASK. Praktyka i Teoria Informacji Naukowej i Technicznej 1993 nr 1 s.19-21.

[LAN76] Langefors B., Samuelson K.: Information in data system. New York 1976.

[LAQ94] Laquey T.: The Internet Companion. A Beginner's Guide to Global Networking. 2nd ed. New York 1994.

[ŁAK95] Łakomy M.: Protokoły TCP/IP. NetWorld Sieci komputerowe i telekomunikacja 1995 nr 7 s.57-60.

[MAT79] Matulka Z.: Selekcja i synteza informacji w procesie samokształcenia. Warszawa 1979.

[MAX94] Maxwell C., Grycz Czesław J.: New Reader's Official Internet Yellow Pages. Indianapolis 1994.

[MGO94] McBride J. S., Godin S.: The Internet White Pages. San Mateo 1994.

[MUR94] Muraszkiewicz M.: Obiektowe bazy danych. Kserokopia bez miejsca wydania 1994.

[MRY94] Muraszkiewicz M., Rybiński H.: Bazy Danych. Warszawa 1993.

[NEW94] Newby G. B.: Directory of Directories on the Internet. A Guide to Information Sources. Westport 1994.

[NIC80] Nichols E. J. : Struktura języków programowania. Warszawa 1980.

[NWJ95] Jak połączyć odległe sieci LAN. NetWorld Sieci komputerowe i telekomunikacja 1995 nr 5 s.26-34.

[PAS95] Podręcznik użytkownika sieci komputerowej. Praca zbiorowa pod red. S. Paszczyńskiego. Warszawa 1995.

[PAC93] Parsaye K., Chignell M.: Intelligent Database Tools & Applications. yperinformation Access, Data Quality, Visualisation, Automatic Discovery. New York 1993.

[PCK89] Parsaye K., Chignell M., Khoshafian S., Wong H.: Intelligent Databases, Object-Oriented, Deductive Hypermedia Technologies. New York [i in.] 1989

[PIW95] Piwowar B.: Jak i kiedy stosować mosty i routery?. NetWorld Sieci komputerowe i telekomunikacja, 1995 nr 6 s.30-42.

[POG81] Pogorzelski W. A.: Klasyczny rachunek kwantyfikatorów, zarys teorii. Warszawa 1981.

[RAS75] Rasiowa H.: Wstęp do matematyki współczesnej. Warszawa 1975.

[QAH89] Qarteman J, H. J.: Notable Computer Networks. Communication of the ACM, Oct. 1989 nr 29, s.932-971.

[RYK94] Rykaczewska-Wiorogórska B.: Usługi i zasoby sieci naukowo-badawczych. Część 1: Poczta elektroniczna. Praktyka i Teoria Informacji Naukowej i Technicznej Cz. 1: Poczta elektroniczna 1994 nr 1(5) s. 13-16 .

[RYB87] Rybiński H.: Modele baz danych. Warszawa 1987

[SCH95] Schatz B.: Building the Interspace: The Illinois Digital Library Project. Communication of the ACM 1995 nr 4 s.62-63

[SAL71] Salton G.: The SMART retrieval system. Experiments in automatic document processing. New Jersey 1971

[SAL77] Yu C. T., Salton G.: Effective inforamation retrieval using term accuracu. Communication of ACM 1977 Nr 3 s. 135-142

[SEA95] Sears J. l.: Harnesing the Wolrd Wide Web. „IEEE Expert" August 1995 s. 42-43.

[SHE95] Sheldon T.: Wielka encyklopedia sieci. Wrocław 1995.

[SIT90] Sitarska A.: Systemowe badanie bibliotek. Studium metodologiczne. Łódź 1990.

[STA87] Stallings U.: Handbook of computer-communication standards vol.1, The open system interconnection (OSI) model and OSI-related standards, New York 1987.

[TAN89] Tanenbaum A. S.: Computer networks vol. 1-3, Prentice-Hall 1989.

[WIL95] Wilensky R.: UC Berkeley's Digital Library Project. Communication of the ACM 1995 nr 4 s. 60

ADRESY W KONWENCJI URL

Bibliografia AI (sztucznej inteligencji)
ftp://ftp.cs.umanitoba.ca/pub/bibiographics/ai/index.html

Babylon (informacje n/t ekspertowego systemu szkieletowego)
ftp://ftp.gmd.de/GMD/ai-research/Software/Babylon/

MInet Machine Learning Archive at GMD
ftp://ftp.gmd.de/ml-archive/README.html

IEEE Expert – spis treści oraz abstrakty
gopher://info.computer.org:TablesofContents/Magazines/IEEEExpert

WebWatcher Front Door
http://webwatcher.learning.cs.cmu.edu:8080/cgi-bin/agent-welcome.pl?
http://www.cs.cmu.edu/Web/FrontDoor.html

SIMS Project
http://www.isi.edu/sims//sims-homepage.html

MIT Media Lab, Autonomous Agents Group
http://les.www.media.mit.edu/groups/agents/

Nobotics Home Page
http://robotics.stanford.edu/groups/nobotics/home.html

The Internet Softbot
http://www.cs.washington.edu/research/projects/softbots/www/softbots.html

Knowledge System Laboratory, National Research Council
http://ai.iit.nrc.ca/home_page.html

Informacje IBIN UW
http://ci.uw.edu.pl/uw/ibin/sss4/pl/sss4.html

Informacje Lockheed AI Center n/t systemu Recon i pozostałych
http://hitchhiker.space.lockheed.com/aic//README.html

Robotics Research in Japan
http://hyp.isk.t.u-tokyo.ac.jp/~tom/rsj.html

HCI Bibliography Project (zawiera informacje bibliograficzne n/t Human Computer Interaction (od 1980 r.)
http://hyperg.ugraz..ac.at:80/D50FFDEC/CHCIbib

Univeristy of Illinois
http://ilg.cs.uiuc.edu/pub/src/

AI-related courses (UK Open University)
http://kmi.open.ac.uk/diploma-msc-info.html

The AI Lab at Hamburg Univeristy
http://lki-www.informatic.uni-hamburg.de/

University of Texas
http://net.cs.utexas.edu/users/ml/

Association for Computing Machinery (ACM)
http://sigart.acm.org/
http://acm.org/

MIT Press
http://www-mitpress.mit.edu/mitp/recent-books/compl/comp-sci-toc.html

American Association for Artificial Intelligence

http://www.aaai.org/

AI Magazine
http://www.aaai.org/Publications/Press/press.html

The Massachusetts Institute of Technology AI Laboratory
http://www.ai.mit.edu/

OFAI and IMKAI library information system (Projekt Wydziałów: Medical Cebernetics, Arificial Intelligence University of Vienna (IMKAI) oraz Austrian Research Institute for Ariticial Intelligence (system zawiera ponad 36.000 dokumentów głównie n/t sztucznej inteligencji)
http://www.ai.univie.ac.at/biblio.html

The Global Campus
http://www.calpoly.edu:80/~delta/

IEEE Computer Society
http://www.computer.org/

Carnegie Mellon Univeeristy Artificial Intelligence Repository
http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/ai-repository/ai/areas/0.html
http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/theo-6web-agent/www/project-home.html

12thMaryland Theory Day, „Man vs. Machine Chess Match" Grandmaster Gennady Segalchik vs. 1800 node intel Paragon Supercomputer Running Socrates
http://www.cs.umbc.edu/conferences/mtd95/mm_match/

The Robotics and Vision Research Group at the University of Western Australia
http://www.cs.uwa.edu.au/robvis/index.html

Lobner Prize Competition in Artificial Intelligence
http://www.csusm.edu/locbner_contest.html

Cambidge university Press Complete On-line Catalog
http://www.cup.cam.ac.uk/Connections/BRS.html

Unniversity Catholique de Louvain Neural Net Group
http://www.dice.ucl.ac.be/neural-nets/NNgroup.html

European Commision
http://www.echo.lu/home.html

System Winterp
http://www.cit.com/software/winterp/winterp.html

Artificial Intelligence: An International Journal (informacje bez abstraktów)
http://www.elsevier.nl/catalogue/SA2/215/08672/505601/505601.html

Univeristy of Californi, Irvine
http://www.ics.uci.edu/AI/ML/MLPrograms.html

Inference Corp (szczegółowe informacje oraz wersja demonstracyjna systemów: ART*Enterprise, CBR2)
http://www.inference.com/
http://www.isi.edu/sims/sims-homepage.html

System Clips
http://www.jsc.nasa.gov/~clips/CLIPS.html

SEL-HPC Article Archive
http://www.lpac.qmw.ac.uk/SEL-HPC/Articles.index.html

National Science Foundation
http://www.nsf.gov/

The Price-Waterhouse Technology Centre, Stany Zjednoczone
http://www.pw.com/

Teknowledge Corporation, informacje n/t programu Intelligent Systems Integration i pozostałych usług.
http://www.teknowledge.com/

The World Lecture Hall
http://www.utexas.edu/world/lecture/

European Coordination Committee for Artificial Intelligence (ECCAI)
http://wwwis.cs.utwente..nl:8080/mars/ECCAI.html

Pedagogic Resources for Teching and Learning Intruductory AI
http://yoda.cis.temple.edu:8080/IIIA/ai.html

WebWatcher tour guide
http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/theo-6/web-agent/www/project-home.html

The FAQFinder project at the University of Chicago
http://cs-www.uchicago/edu/burke/faqfinder.html

Sims, an information mediator being developed at the Information Science Institute
http://www.isi.edu/sims/sims-homepage.html

The Autonomus Agent Group at MIT´s Media Lab:
http://agents.www.media.mit.edu/goups/agents/

The Nobotics Research Group at Stanford University
http://robotics.stanford.edu/groups/nobotics/home.html

The Internet Softbot project at the University of Washington
http://www.cs.washington.edu/research/softbots

OZNACZENIA I SKRÓTY

Symbole matematyczne

2^A – rodzina (zbiór) wszystkich podzbiorów zbioru A (zbiór potęgowy zbioru A)

A x B – produkt (iloczyn) kartezjański zbioru AxB

$\rho \subseteq$ **A x B** – relacja dwuargumentowa na zbiorach A x B

x ρ y – elementy x, y spełniające relację ρ

$<a_1,\ldots,a_n>$ – ciąg a1,…,an

$\alpha: A \xrightarrow{w} B$ – funkcja z A w B

$\alpha: A \xrightarrow[na]{} B$ – funkcja z A na B

$\alpha: A \xrightarrow[1\text{-}1]{} B$ – funkcja różnowartościowa

$\alpha: A \xrightarrow[n\text{-}1]{} B$ – funkcja wielowartościowa

$\alpha: A \xrightarrow[n\text{-}m]{} B$ – odwzorowanie

α^{-1} – funkcja odwrotna do α

∀ – kwantyfikator ogólny (dla każdego)

∃ – kwantyfikator szczegółowy (istnieje)

∪ – operator sumowania zbiorów (dwuargumentowy)

∩ – operator koniunkcji zbiorów (dwuargumentowy)

\ – operator uzupełnienia zbioru (jednoargumentowy)

Oznaczenia w tekście

■ – koniec definicji

▲ – koniec przykładu

◆ – koniec twierdzenia i dowodu

Ważniejsze skróty

CCIT	International Telegraph and Telephone Consultative Committee
CIUW	Centrum Informatyczne Uniwersytetu Warszawskiego
CREN	Corporation for Research and Educational Networking
DNS	Domain Name System
FAQ	Frequently Asked Questions
FDDI	Fiber Distributed Data Interface

FTAM	File Transfer, Access oraz Management
GUI	Grafical User Interface
IBIN UW	Instytut Bibliotekoznawstwa i Informacji Naukowej
IEEE	Institute of Electronic and Electrical Engineers
ISO/OSI	International Standards Organization Open Systems Interconnection Reference Model
JTM	Job Transfer and Manipulation
LAN	Local area network
MAN	Metropolitan area network
MAP	Manufacturing Automation Protocol
NASK	Naukowa i Akademicka Sieć Komputerowa
NSF	National Science Foundation
PPP	Point to Point Protocol
SNA	System Network Architecture
SLIP	Serial Link Internet Protocol
SZDB	System Zarządzania Bazą Danych
TOP	Technical and Office Protocol
UUCP	Unix to Unix CoPy
VTP	Virtual Terminal Protocol
WAN	Wide Area Network

SKOROWIDZ