

Jacek FRACZEK, Jacek KMONK, Stanisław KOZIELSKI,  
Dorota PIERZCHAŁA, Robert TUTAJEWICZ  
Politechnika Śląska, Instytut Informatyki

## INTEGRACJA BAZ DANYCH - PRZEGLĄD METOD I NARZĘDZI; ANALIZA POTRZEB NA WYBRANYM PRZYKŁADZIE<sup>1</sup>

**Streszczenie.** W pracy omówiono metody i narzędzia integracji baz danych. Przedstawiono klasyfikację poziomów integracji, rozróżniając rozproszone bazy danych oraz kilka wariantów systemów wielobazowych. Wskazano na możliwość integracji baz danych poprzez budowę hurtowni danych. Przeprowadzono analizę potrzeb w zakresie integracji systemów użytkowych Urzędu Miasta. Przedstawiono koncepcję uniwersalnej aplikacji zapewniającej dostęp do lokalnych baz danych.

## DATABASES INTEGRATION - METHODS AND TOOLS REVIEW; THE ANALYSIS OF NEEDS OF A CHOSEN EXAMPLE

**Summary.** The paper describes the methods and tools of databases integration. It discusses the classification of integration levels and differentiates distributed databases and several variants of multidatabase systems. The possibility of databases integration by use of data warehouses is indicated. The paper also contains the analysis of needs of town office utility systems integration. The concept of an universal application that assures access to local databases is discussed.

### 1. Wstęp

Lawinowy przyrost liczby systemów informatycznych wspomagających różne dziedziny ludzkiej działalności rodzi z biegiem czasu oczywistą potrzebę zapewnienia współdziałania takich systemów. Znaczna część programów użytkowych korzysta z baz danych, stąd też

---

<sup>1</sup> Praca wykonana w ramach Projektu Celowego nr 8T11C 009 96C/2995

problem współdziałania prowadzi do zagadnień integracji baz danych. Konieczność integracji staje się nieodzowna, jeśli powstaje koncepcja stworzenia globalnego systemu zarządzania przedsiębiorstwem przy zachowaniu (i włączeniu do takiego systemu) dotychczas eksploatowanych systemów użytkowych.

W niniejszym opracowaniu dokonano przeglądu metod integracji baz danych, główną uwagę skupiono przy tym na analizie własności wybranych narzędzi umożliwiających programom użytkowym dostęp do wielu baz danych.

W punkcie 2 przedstawiono klasyfikację i taksonomię poziomu integracji systemów wykorzystujących wspólne dane. Jako podstawę takiej oceny przyjęto bliskość powiązań integrowanych systemów. Obok rozwiązania znanego jako tzw. rozproszone bazy danych scharakteryzowano różne warianty tzw. systemów wielobazowych.

Punkt 3 poświęcono prezentacji najnowszego podejścia do integracji, opartego na modelu tzw. hurtowni danych. Rozwiązania wykorzystujące hurtownię danych dają wygodną podstawę do budowy systemów zarządzania całym przedsiębiorstwem, systemów wspomaganie decyzji oraz specjalizowanych systemów zaawansowanej i wielowymiarowej analizy danych.

W punkcie 4 przedstawiono wybrane narzędzia programowe umożliwiające integrację baz danych. Szczególną uwagę zwrócono na technikę ODBC oraz narzędzia typu router czy gateway. Przedstawiono też sposoby integracji różnych baz danych przy użyciu generatora aplikacji MAGIC.

W punkcie 5 przeprowadzono analizę potrzeb zgłaszanych przez użytkowników w zakresie integracji systemów informatycznych Urzędu Miasta. Omówiono koncepcję aplikacji zapewniającej wielu programom użytkowym dostęp do lokalnej bazy danych systemu Ewidencji Ludności.

## 2. Poziomy integracji systemów wykorzystujących wspólne dane

W niniejszym rozdziale dokonano krótkiego przeglądu (w oparciu o [1]) rodzajów powiązań systemów zarządzania bazami danych. Wśród wyróżnionych poziomów integracji skrajne pozycje zajmują: systemy rozproszonych baz danych oraz bardzo ogólnie rozumiane systemy współdziałające. Rozwiązania pośrednie znane są jako tzw. systemy wielobazowe.

Ogólny termin "system wielobazowy" [1, 6] obejmuje takie klasy, jak: systemy baz danych o schemacie globalnym, sfederowane bazy danych, systemy wielobazowe powiązane na poziomie języka zapytań oraz homogeniczne systemy wielobazowe powiązane na poziomie języka zapytań. System wielobazowy jest systemem rozproszonym, który funkcjonuje jako pośrednictwo (ang. *front-end*) do wielu lokalnych Systemów Zarządzania Bazami Danych (SZBD) lub też jest umiejscowiony w strukturze jako warstwa systemu globalnego na szczycie lokalnego SZBD. System globalny dostarcza wszystkich funkcji bazy danych

i współdziała z lokalnymi SZBD w ich zewnętrznym interfejsie użytkownika. Chociaż każdy lokalny węzeł musi obsługiwać pewne globalne funkcje interfejsu dla systemu globalnego, to lokalne SZBD są autonomiczne. System globalny dostarcza pewnych środków (globalny schemat lub język systemu wielobazowego) dla rozwiązywania problemów różnic w reprezentacji danych oraz różnic w funkcjach pomiędzy lokalnymi SZBD. Założenie takiej możliwości jest konieczne, ponieważ ta sama informacja może być utrzymywana w wielu miejscach w różnej formie. Użytkownik globalny winien mieć dostęp do informacji pochodzącej z wielu źródeł w jednakowy, dość prosty sposób.

## 2.1. Rozproszone bazy danych

Rozproszone bazy danych (ang. *distributed databases*) są najściślej połączonymi systemami o rozproszonej informacji. Łączone lokalne SZBD są zwykle homogeniczne (w odniesieniu do zaimplementowanego modelu danych) i prezentują te same funkcjonalne interfejsy, nawet jeśli mogą być implementowane na różnych maszynach.

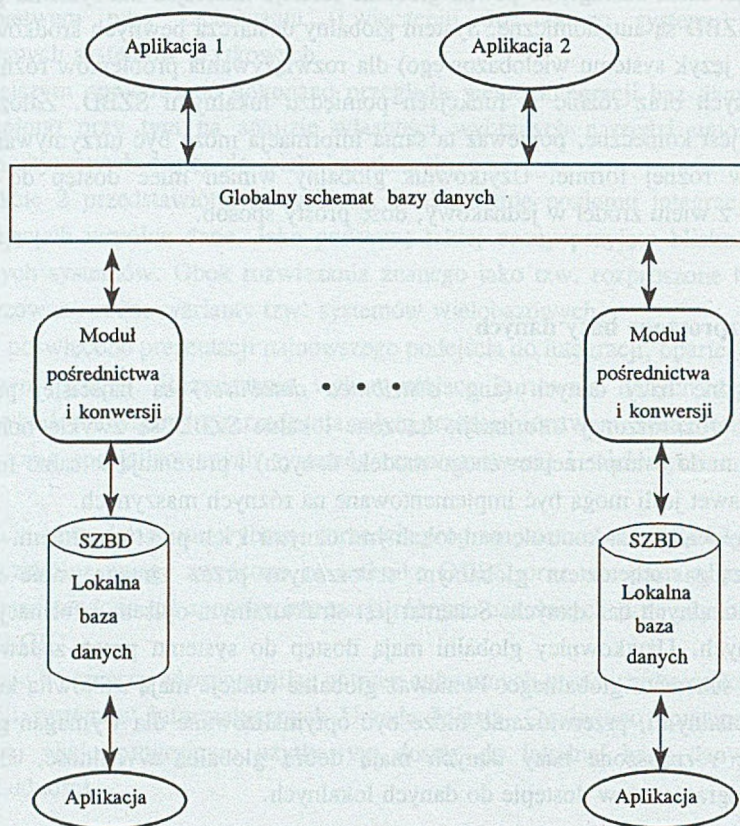
System globalny ma kontrolę nad lokalnymi danymi i ich przetwarzaniem. System ten zwykle zarządza schematem globalnym stworzonym przez zintegrowanie schematów wszystkich lokalnych baz danych. Schemat jest strukturalnym opisem informacji dostępnej w bazie danych. Użytkownicy globalni mają dostęp do systemu przez zadawanie pytań dotyczących schematu globalnego. Ponieważ globalne funkcje mają całkowitą kontrolę nad funkcjami lokalnymi, przetwarzanie może być optymalizowane dla wymagań globalnych. W rezultacie rozproszone bazy danych mają dobrą globalną wydajność, ale kosztem znaczących ograniczeń w dostępie do danych lokalnych.

## 2.2. Systemy wielobazowe o schemacie globalnym

Systemy wielobazowe o schemacie globalnym (ang. *global schema multidatabases*) są luźniej połączone niż rozproszone bazy danych, ponieważ funkcje globalne mają dostęp do lokalnych informacji poprzez zewnętrzny interfejs użytkownika lokalnego SZBD. W tym rozwiązaniu tworzony jest globalny schemat, którego utrzymanie wymaga ścisłej współpracy systemów lokalnych.

Systemy wielobazowe o schemacie globalnym mogą integrować wcześniej istniejące lokalne SZBD bez ich modyfikowania. Zwykle integrują one też heterogeniczne lokalne SZBD. Heterogeniczność ta może obejmować różne modele danych lub różne implementacje tego samego modelu. Dlatego tworzenie globalnego schematu jest znacznie trudniejszym problemem w systemach wielobazowych niż w systemach rozproszonych baz danych, gdzie lokalne SZBD są zwykle homogeniczne.

Ogólną strukturę omawianych systemów przedstawia rys. 1.



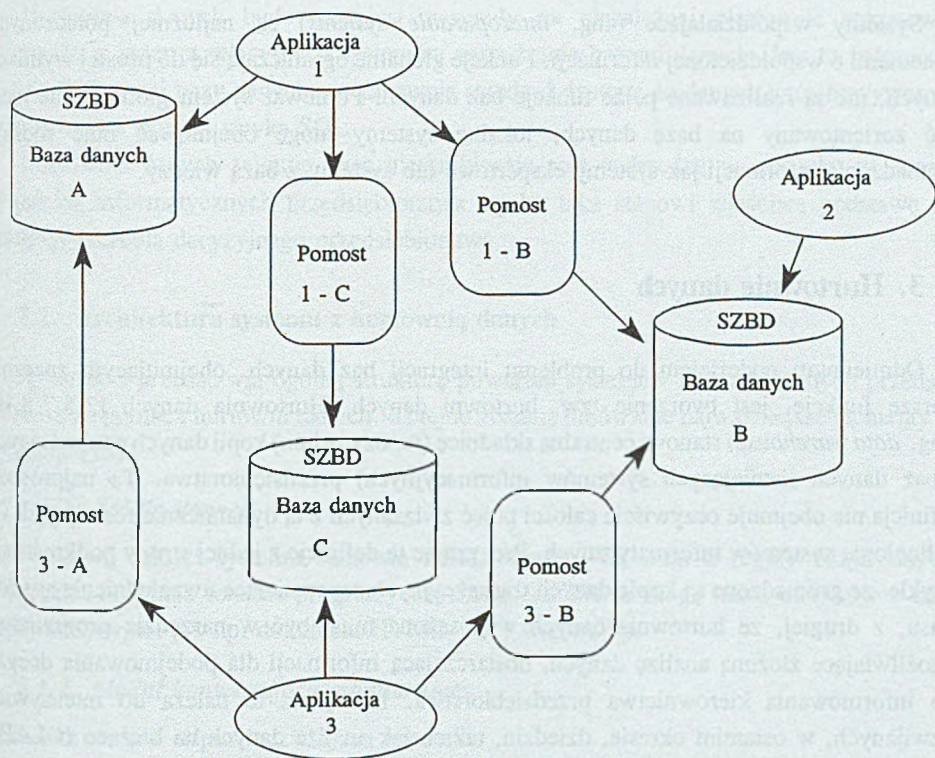
Rys. 1. Ogólna struktura systemów wielobazowych o schemacie globalnym  
 Fig. 1. General structure of global schema multidatabases

### 2.3. Sfederowane bazy danych

Sfederowane bazy danych (ang. *federated databases*) są luźniej połączonym podzbiorem systemów wielobazowych o schemacie globalnym. W tym rozwiązaniu nie istnieje pojedynczy schemat globalny. Każdy lokalny system zawiera lokalny schemat dla funkcji eksportu i lokalny schemat dla funkcji importu. Schemat dla funkcji eksportu jest opisem informacji, którą lokalny węzeł współdzieli z systemem globalnym. Schemat dla funkcji importu jest opisem informacji (zarówno reprezentacji danych, jak i pochodzenia danych) w odległych węzłach, które mogą być dostępne w danym węźle. Pytania użytkownika są ograniczone do danych lokalnych oraz danych reprezentowanych w lokalnym schemacie dla funkcji importu.

## 2.4. Systemy wielobazowe powiązane na poziomie języka zapytań

Takie systemy (ang. *multidatabase language systems*) są luźniej połączone od poprzedniej klasy, ponieważ żaden (nawet częściowy) schemat globalny nie jest utrzymywany. System globalny realizuje funkcje globalnej bazy danych przez udostępnienie narzędzi języka zapytań do integrowania informacji z oddzielnych baz danych. Pytania użytkownika mogą specyfikować dane w lokalnym schemacie każdego węzła należącego do systemu. Podobnie jak systemy o schemacie globalnym, omawiane systemy integrują już istniejące, heterogeniczne, lokalne SZBD bez modyfikowania ich. Przykładami narzędzi programowych zapewniających dostęp do oddzielnych baz danych są pomosty (routery, gatewaye, sterowniki ODBC). Ten rodzaj powiązań ilustruje rys. 2.



Rys. 2. Systemy wielobazowe powiązane za pomocą pomostów na poziomie języka zapytań  
Fig. 2. Multidatabase language systems connected by means of gateways

## 2.5. Homogeniczne systemy wielobazowe powiązane na poziomie języka zapytań

Systemy te (ang. *homogenous multidatabase language systems*) są oddzielną formą systemów wielobazowych powiązanych na poziomie języka zapytań. Stanowią one odrębną klasę, ponieważ istnieje szereg projektów systemów wielobazowych, które aktualnie oparte są tylko na homogenicznych lokalnych SZBD. Należą do nich pierwsze komercyjne systemy wielobazowe. W rozwiązaniach komercyjnych daje się zauważyć ograniczenie funkcji języka w porównaniu z projektami poprzednio omówionej klasy. Systemy te w równym stopniu ujawniają cechy rozproszonych baz danych jak i systemów wielobazowych.

## 2.6. Systemy współdziałające

Systemy współdziałające (ang. *interoperable systems*) są najluźniej połączonymi systemami o współdzielonej informacji. Funkcje globalne ograniczają się do prostej wymiany danych; nie są realizowane pełne funkcje baz danych. Ponieważ system globalny nie musi być zorientowany na bazę danych, lokalne systemy mogą obejmować inne rodzaje gromadzenia informacji jak systemy ekspertowe lub systemy z bazą wiedzy.

# 3. Hurtownie danych

Odmiernym podejściem do problemu integracji baz danych, obejmującym znacznie szersze funkcje, jest tworzenie tzw. hurtowni danych. Hurtownia danych [2,5,7,8,10] (ang. *data warehouse*) stanowi centralną składnicę (repozytorium) kopii danych pochodzących z baz danych (istniejących systemów informacyjnych) przedsiębiorstwa. Ta najprostsza definicja nie obejmuje oczywiście całości pojęć związanych z tą dynamicznie rozwijającą się technologią systemów informatycznych. Precyzując tę definicję z jednej strony podkreśla się zwykle, że gromadzone są kopie danych transakcyjnych zapewniające uwzględnienie aspektu czasu, z drugiej, że hurtownia danych wyposażona musi być w narzędzia programowe umożliwiające złożoną analizę danych, dostarczającą informacji dla podejmowania decyzji lub informowania kierownictwa przedsiębiorstwa. Narzędzia te należą do intensywnie rozwijanych, w ostatnim okresie, dziedzin, takich jak analiza danych na bieżąco (OLAP - ang. *On Line Analytical Processing*) czy eksploracja (lub drążenie) danych (ang. *data mining*).

Z uwagi na cele niniejszego opracowania istotny jest fakt, że do hurtowni danych dostarczane są dane z wszystkich systemów informacyjnych (baz danych) przedsiębiorstwa. W tym ujęciu stanowi więc ona naturalny element integracji baz danych.

Do najważniejszych cech hurtowni danych zaliczamy [2]:

- Tematyczne zorientowanie: dane są zorganizowane zgodnie z problemem analiz. Dane zorganizowane tematycznie zawierają tylko informacje niezbędne dla procesu podejmowania decyzji.
- Integracja: sposób kodowania podobnych danych przechowywanych w wielu oddzielnych aplikacjach jest często niespójny. W trakcie kopiowania tych danych do hurtowni danych następuje konwersja ujednocniająca postać danych.
- Zmienność czasowa: Hurtownia danych może zawierać dane z kilku czy kilkunastu lat, dla umożliwienia analizy porównawczej, oceny trendów oraz tworzenia prognoz. Dane te nie są aktualizowane.
- Nieulotność: Dane umieszczone w hurtowni nie są aktualizowane ani zmieniane, są one tylko udostępniane do odczytu.

Hurtownia danych budowana jest z reguły na niezależnej platformie sprzętowej, w oparciu o jeden z relacyjnych systemów zarządzania bazami danych. Jest to najczęściej serwer SQL-owej bazy danych, co zapewnia standard dostępu do danych przechowywanych w hurtowni w postaci języka SQL.

Hurtownia danych zajmuje więc hierarchicznie rolę nadrzędną w stosunku do innych systemów informatycznych przedsiębiorstwa i jako taka stanowi właściwą podstawę dla aplikacji szczebla decyzyjnego przedsiębiorstwa.

### 3.1. Architektura systemu z hurtownią danych

Rysunek 3 przedstawia ogólną strukturę powiązań systemów informatycznych przedsiębiorstwa za pomocą hurtowni danych. Kolejno zostaną omówione najważniejsze elementy tej struktury [15].

#### 3.1.1. Źródła danych

Podstawę całości systemu stanowią źródła danych. Są nimi z reguły bazy danych, z których korzystają lokalne programy użytkowe. Źródła te mogą także obejmować dane mniej tradycyjne, jak np. dokument HTML, bazy wiedzy itp.

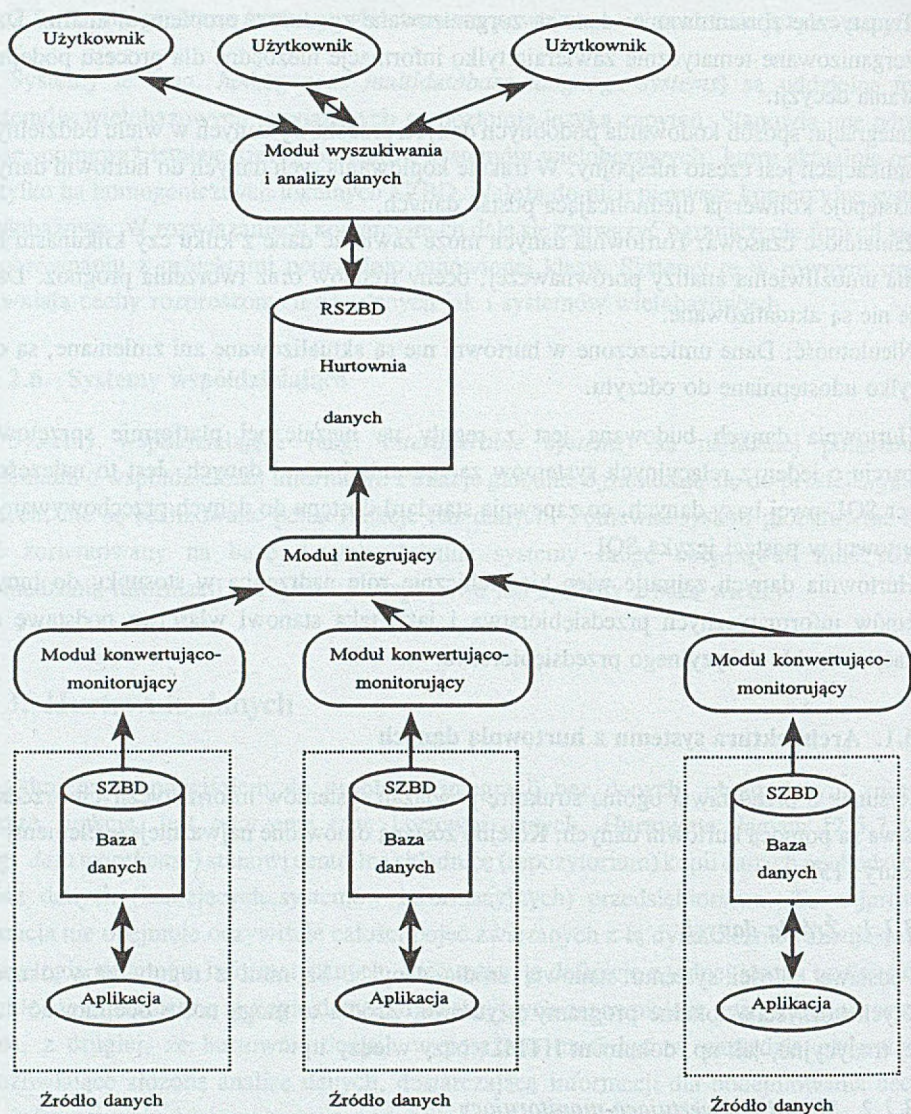
#### 3.1.2. Moduł konwertująco-monitorujący

Element ten jest odpowiedzialny za operacje:

- Translacja: przekształcanie danych z postaci przechowywanej w źródle danych (np. płaskie pliki) do postaci w hurtowni (z reguły relacyjnej).
- Detekcja zmian: wykrycie faktu zmian wartości danych i przekazanie tych zmian do integratora.

Rozważenie problemu detekcji prowadzi do wyróżnienia kilku typów źródeł informacji [15]:

- 1) Źródła kooperujące: dysponują triggerami lub innymi mechanizmami aktywnych baz danych. Daje to możliwość automatycznej rejestracji zmian danych.



Rys. 3. Architektura systemu z hurtownią danych  
 Fig. 3. Architecture of the system with a data warehouse

- 2) Źródła dziennikowane (ang. *logged*): prowadzony jest dziennik transakcji dla źródeł, co daje możliwość analizy i wyodrębniania wszystkich zmian.
- 3) Źródła odpytywalne (ang. *queryable*): źródła te umożliwiają modułowi konwertująco-monitorującemu wyszukiwanie (odpytywanie) informacji. Okresowa analiza danych pozwala wtedy wykryć zmiany danych.



4) Źródła migawkowe (ang. *snapshot*): dla źródeł, które nie posiadają własności 1-3, wykonywane są okresowe zrzuty (migawkowe obrazy). Porównanie tych migawek daje możliwość wykrycia zmian.

### 3.1.3. Integrator

Zadaniem integratora jest przyjmowanie zmian danych przesyłanych z modułu konwertująco-monitorującego i odnotowywanie tych zmian w zawartości hurtowni.

Na odpowiednim poziomie abstrakcji dane w hurtowni możemy interpretować jako zmaterializowane perspektywy. W takim ujęciu zadaniem integratora jest utrzymywanie tych perspektyw.

### 3.1.4. Moduł wyszukiwania i analizy danych

Moduł wyszukiwania i analizy danych realizuje złożone zadania, określone swą nazwą, dla specyficznych użytkowników końcowych. Są tu zazwyczaj wykorzystywane narzędzia zaliczane do klasy systemów OLAP oraz *data mining*.

## 3.2. Podsumowanie cech hurtowni danych

Kluczem koncepcji hurtowni danych jest prowadzenie wyboru, filtracji i integracji danych w sposób wyprzedzający zadawanie pytań (zadań wyszukiwania i analizy danych) przez użytkowników. Kiedy pytania takie są już zadawane, nie muszą być tłumaczone i przesyłane do źródeł danych dla ich wykonania (jak to ma miejsce np. w systemach wielobazowych lub w rozproszonych bazach danych). Takie tłumaczenie i przesyłanie pytań może być złożone i czasochłonne, zwłaszcza jeśli pytanie obejmuje wiele źródeł danych. Dlatego koncepcja hurtowni danych może być uważana za aktywne podejście do integracji informacji w porównaniu do tradycyjnego, pasywnego podejścia, gdzie przetwarzanie i integrowanie rozpoczyna się w momencie pojawiania się pytania. Hurtownia danych stwarza też bezpieczny i długoterminowy magazyn dla ważnych danych, będący w dodatku pod kontrolą użytkowników hurtowni.

Główną potencjalną wadą modelu hurtowni danych jest gromadzenie kopii danych z oryginalnych źródeł, co zajmuje dodatkową, zwykle dużą, pamięć. Jednakże fakt spadku cen pamięci z jednej strony, oraz filtracja i agregatyzacja danych przed wprowadzeniem ich do hurtowni z drugiej strony, sprawiają, że problem ten nie jest w istocie tak ważny. Poważniejszym zagadnieniem jest natomiast fakt, że kopiowanie i przetwarzanie danych wprowadza opóźnienia sprawiające, że dane w hurtowni zwykle nie są całkiem aktualne. Ponadto administrator hurtowni, decydując o sposobie wyboru i agregatyzacji danych, musi przewidywać wszystkie zadania, jakie postawią przed hurtownią użytkownicy. Jeśli stawiane zadania wykraczają poza zakres bądź sposób gromadzenia danych, lub też konieczność dostępu do najświeższych danych jest elementem krytycznym, wtedy należy udostępnić

użytkownikom dodatkowe narzędzia umożliwiające bezpośredni dostęp do źródeł danych poza hurtownią.

## 4. Omówienie wybranych narzędzi programowych umożliwiających integrację baz danych

Istnieje obecnie znaczna liczba narzędzi programowych ułatwiających szeroko rozumianą integrację baz danych. Typowym zadaniem, wykonywanym za pomocą tych narzędzi jest dostęp w jednej aplikacji (programie użytkowym) do wielu baz danych, utworzonych za pomocą różnych SZBD. Narzędzia te pozwalają więc tworzyć systemy wielobazowe powiązane na poziomie języka zapytań. Kolejno omówione zostaną trzy różne klasy tych narzędzi.

### 4.1. Standard ODBC

Open Database Connectivity (ODBC) jest standardem dostępu do danych rozmieszczonych na różnych serwerach baz danych bądź serwerach plików. Aplikacja operująca na takich źródłach danych nie jest więc uzależniona od interfejsu specyficznego dla danego systemu zarządzania bazą danych. Jest to oczywiście możliwe przy zachowaniu odpowiedniej struktury tabel. Standard ten jest dostępny w środowisku MS Windows.

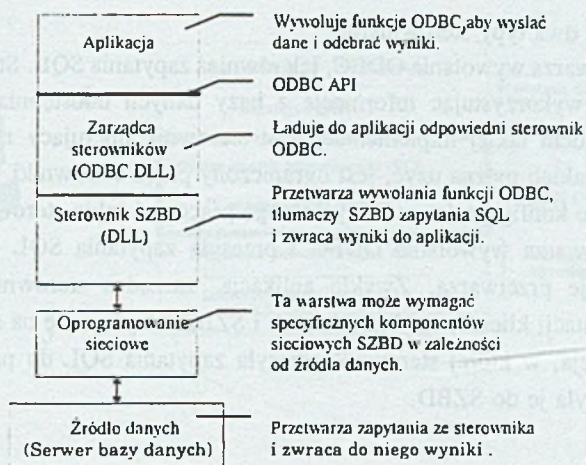
ODBC definiuje wspólny interfejs API (ang. *Application Programming Interface*), dzięki któremu rozwiązuje problem łączenia się z różnymi bazami danych. Każda aplikacja używa tego samego kodu (zdefiniowanego przez specyfikację API) do porozumiewania się z różnymi typami źródeł danych za pomocą sterownika specyficznego dla danego SZBD. Sterowniki te są zaimplementowane jako biblioteki DLL (ang. *dynamic-link library*).

#### 4.1.1. Architektura ODBC

W architekturze ODBC (rys. 4) można wyróżnić cztery składniki:

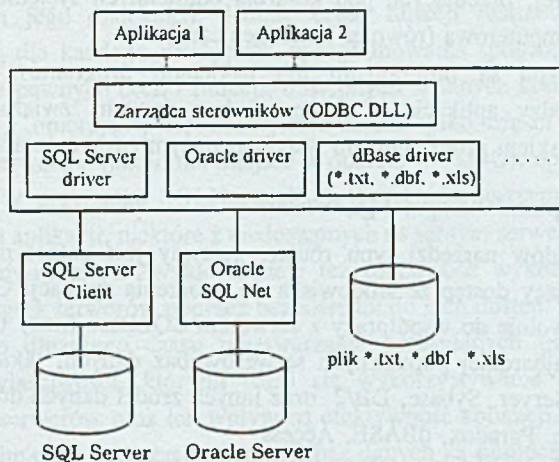
- aplikację - wywołuje funkcje ODBC API w celu połączenia się ze źródłem danych, wysłania i odebrania danych oraz zerwania połączenia, reaguje na błędy ODBC,
- zarządcę sterowników - (ang. Driver Manager) dostarcza aplikacji informacje o dostępnych źródłach danych; łąduje do aplikacji odpowiedni sterownik ODBC,
- sterownik - przetwarza wywołania funkcji ODBC, zarządza wymianą danych między aplikacją a SZBD, tłumaczy formaty danych, tłumaczy standardową składnię SQL do SQL'a rodzimego dla danego źródła danych, zamienia kody błędów ze źródeł danych na kody błędów według standardu ODBC, jeśli zachodzi potrzeba, to deklaruje i manipuluje kursorami, oraz inicjalizuje transakcje (co nie jest widoczne dla aplikacji),
- źródło danych - zawiera dane, do których użytkownik chce mieć dostęp oraz udostępnia metody umożliwiające ten dostęp.

Rys. 5 pokazuje strukturę powiązań między tymi czterema składnikami.



Rys. 4. Architektura ODBC

Fig. 4. ODBC architecture



Rys. 5. Dostęp do różnych serwerów baz danych

Fig. 5. Access to various database servers

Jeśli użytkownik chce skorzystać z danych, umieszczonych na innego typu serwerze baz danych, musi jedynie zainstalować sterownik ODBC odpowiedni dla tego serwera.

#### 4.1.2. Typy sterowników

ODBC definiuje dwa typy sterowników:

- Sterownik przetwarza wywołania ODBC, jak również zapytania SQL. Sterownik realizuje zapytania SQL wykorzystując informacje z bazy danych udostępniane przez serwer plików. Przykładem takiej implementacji jest sterownik operujący na plikach xBase. Zbiór zapytań, jakich można użyć, jest ograniczony przez sterowniki. Na rys. 5 przedstawione są dwie konfiguracje, w których mogą pracować takie sterowniki.
- Sterownik przetwarza wywołania ODBC i przesyła zapytania SQL do serwera bazy danych, który je przetwarza. Zwykle aplikacja, zarządca sterowników i sterownik znajdują się na stacji klienta, zaś baza danych i SZBD znajdują się na serwerze. Istnieje także konfiguracja, w której sterownik przesyła zapytania SQL do procesu gatewaya, a gateway przesyła je do SZBD.

#### 4.2. Narzędzia typu router, gateway

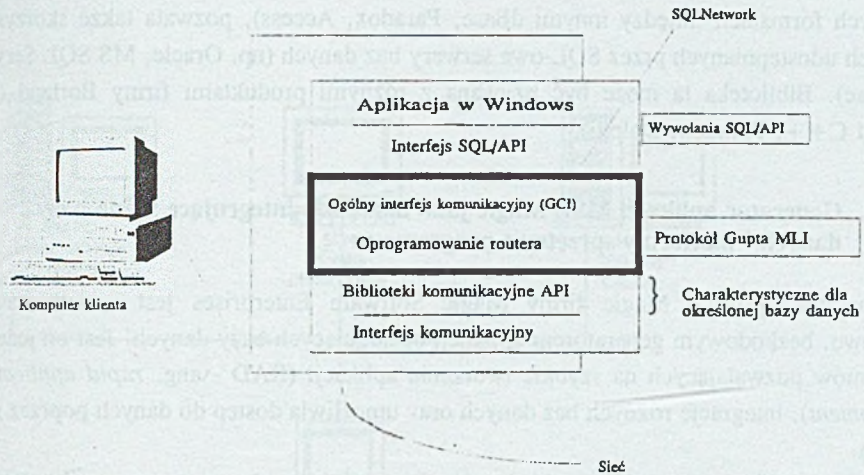
Integracja baz danych za pomocą narzędzi typu router, gateway polega na umożliwieniu korzystania z informacji zgromadzonych na różnych serwerach baz danych, przez określoną aplikację, w sposób maksymalnie przezroczysty. Integrowane serwery baz danych, pochodzące od różnych producentów, mogą być zainstalowane na komputerach o różnej architekturze sprzętowej, pracujących pod kontrolą odmiennych systemów operacyjnych, połączonych siecią komputerową (również heterogeniczną).

Narzędzia tego typu są programami lub pakietami programów pośredniczącymi w komunikacji pomiędzy aplikacją a serwerem baz danych, związanymi najczęściej z określonym środowiskiem programowym wspomagającym tworzenie aplikacji.

##### 4.2.1. Pakiet programów Gupta SQLNetwork

Jednym z przykładów narzędzi typu router, gateway jest pakiet programów Gupta SQLNetwork, realizujący dostęp ze środowiska do tworzenia aplikacji Gupta SQLSystem (przeznaczonego pierwotnie do współpracy z serwerem SQLBase firmy Centura Software, dawniej Gupta) do najbardziej popularnych serwerów baz danych, takich jak: Informix, Ingres, Oracle, SQL Server, Sybase, DB/2 oraz innych źródeł danych dostępnych poprzez drajwery ODBC, m.in. Paradox, dBASE, Access.

Pakiet SQLNetwork składa się głównie z bibliotek specjalizowanych dla każdego integrowanego źródła danych, które korzystają z oprogramowania komunikacyjnego charakterystycznego dla każdego z serwerów (lub z drajwerów ODBC) w celu uzyskania dostępu do baz danych na tych serwerach poprzez sieć oraz w celu uniezależnienia oprogramowania klienta od protokołów sieciowych i platformy sprzętowej komputerów będących serwerami baz danych (rys. 6).



Rys. 6. Oprogramowanie stacji klienta korzystającego z SQLNetwork  
 Fig. 6. Client software for a SQLNetwork workstation

Serwery baz danych, które można zintegrować za pomocą pakietu programów Gupta SQLNetwork, posiadają wiele wspólnych cech i podobnych funkcji; językiem manipulowania danymi jest w każdym przypadku język SQL, z niewielkimi odstępstwami od standardu w poszczególnych jego dialektach. Jednak część funkcji realizowana jest w odrębny, charakterystyczny, dla każdego producenta oprogramowania sposób, niektóre serwery nie posiadają w ogóle pewnych cech i funkcji, dostępnych w innych systemach.

Przy integracji opierającej się na wykorzystaniu tylko części wspólnej możliwości poszczególnych serwerów (jak to ma miejsce w oparciu o ODBC), oprogramowanie traci na funkcjonalności. W przypadku SQLNetwork dla uzyskania maksymalnej przezroczystości z punktu widzenia aplikacji, niektóre z niedostępnych na samym serwerze funkcji emulowane są przez programy-routery. Zwykle istnieje też możliwość wykorzystania oryginalnych możliwości każdego z serwerów poprzez bezpośredni do nich dostęp. Ponieważ odbywać się to może kosztem dłuższego czasu przetwarzania przesyłanych informacji, programista powinien znać właściwości, którymi różni się wykorzystywanie danych pochodzących z integrowanych serwerów oraz ich wpływ na efektywność aplikacji.

Szczegóły techniczne dotyczące integracji baz danych za pomocą pakietu SQLNetwork oraz porównanie czasów realizacji podstawowych operacji na danych rozmieszczonych na różnych serwerach (Ingres, Informix, SQLBase i SQL Server) zawierają artykuły [12, 13, 14].

#### 4.2.2. Biblioteka BDE

Przykładem innego rozwiązania firmowego, umożliwiającego integrację baz danych, jest biblioteka Borland Database Engine (BDE) oferowana przez firmę Borland wraz z większością

cią jej produktów. Umożliwia ona dostęp do wielu baz danych zapisanych w plikach o różnych formatach (między innymi dBase, Paradox, Access), pozwala także skorzystać z danych udostępnianych przez SQL-owe serwery baz danych (np. Oracle, MS SQL Server, Interbase). Biblioteka ta może być używana z różnymi produktami firmy Borland (np. Borland C++, Paradox, Delphi).

#### 4.3. Generator aplikacji MSE Magic jako narzędzie integrujące różne bazy danych i platformy sprzętowo-programowe

Generator aplikacji Magic firmy Magic Software Enterprises jest zorientowanym obiektowo, bezkodowym generatorem aplikacji obsługujących bazy danych. Jest on jednym z systemów pozwalających na szybkie tworzenie aplikacji (RAD - ang. *rapid application development*), integrację różnych baz danych oraz umożliwia dostęp do danych poprzez sieć Internet.

Generator Magic stanowi przykład wizualnego środowiska programowania. Proces tworzenia aplikacji magic'owych wspomagany jest przez generator, a idea tego procesu oparta jest na wykorzystywaniu tabel, wypełnianych i modyfikowanych przez programistę (stąd Magic jest generatorem bezkodowym). Magic pozwala ponadto na tworzenie aplikacji pracujących w architekturze klient-serwer.

Aplikacje Magic'a są całkowicie przenaszalne pomiędzy popularnymi systemami operacyjnymi, takimi jak: DOS, Windows NT/95/3.11, Sun Solaris, HP/UX, AS/400, Open VMS.

Generator Magic umożliwia jednoczesną pracę z wieloma systemami baz danych, zlokalizowanymi na różnych platformach sprzętowych. Dostęp do baz danych realizowany jest poprzez specjalne moduły programowe zwane gatewayami (*MagicGate*, rys. 7). Wśród obsługiwanych przez generator systemów baz danych można wyróżnić: Btrieve, xBase, Oracle, Ingres, Informix, MS SQL, Sybase, C-ISAM, Rdb, RMS, DB2/6000, DB2/400.

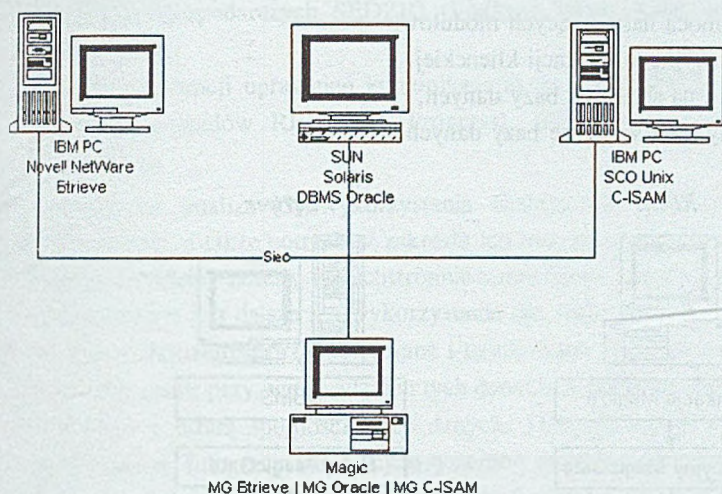
W środowisku generatora Magic wykorzystywane są dwa rodzaje gateway'ów:

- 1) pełny gateway,
- 2) gateway definicyjny.

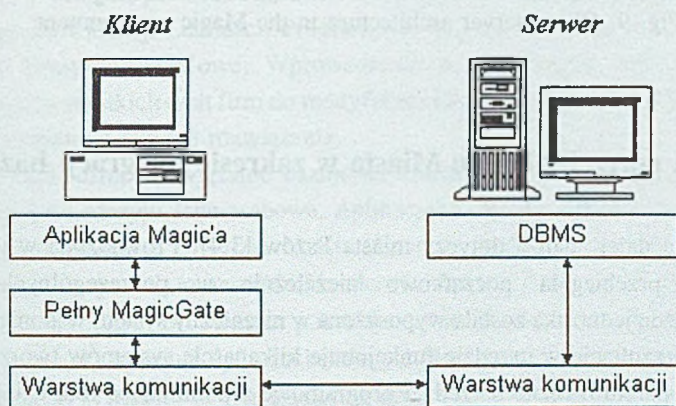
W przypadku pracy z lokalną bazą danych stosuje się pełny gateway, natomiast w przypadku pracy z bazami odległymi możliwe są dwa modele dostępu do bazy danych oparte na:

- 1) "standardowej" architekturze klient-serwer - realizowanej w oparciu o serwer bazy danych (SZBD),
- 2) "magicowej" architekturze klient-serwer - realizowanej w oparciu o oprogramowanie serwera Magic'a.

W przypadku pracy z SZBD realizującym usługi klient-serwer Magic potrafi wykorzystać możliwości SZBD wynikające z pracy w takiej architekturze. Standardowa architektura klient-serwer realizowana jest w oparciu o pełny gateway uruchamiany na stacji klienckiej (rys. 8). Konfiguracja ta, oprócz gateway'a i samego generatora, nie wymaga stosowania innych, dodatkowych modułów.



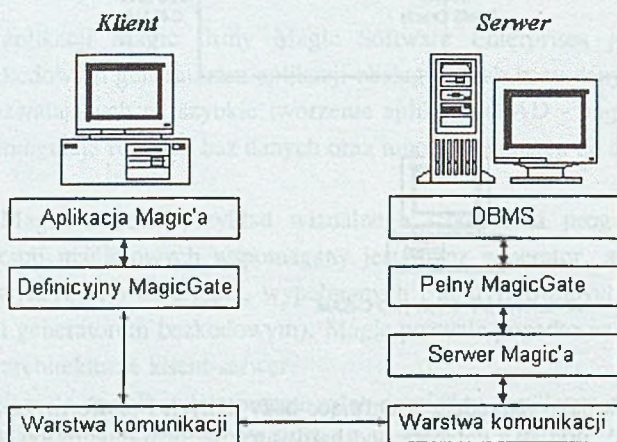
Rys. 7. Magic jako narzędzie integrujące bazy danych i platformy sprzętowe  
 Fig. 7. Magic as a database and hardware platform integration tool



Rys. 8. Magic w standardowej architekturze klient/serwer  
 Fig. 8. Magic in standard client-server architecture

Magic'owa architektura klient-serwer (rys. 9) może być stosowana wszędzie tam, gdzie nie można jej zrealizować na poziomie SZBD. W konfiguracji tej Magic uzyskuje dostęp do danych poprzez serwer Magic'a zainstalowany na serwerze bazy danych. Ten model pracy realizowany jest za pomocą następujących modułów:

- 1) gateway'a definicyjnego na stacji klienckiej,
- 2) serwera Magic'a na serwerze bazy danych,
- 3) pełnego gateway'a na serwerze bazy danych.



Rys. 9. Architektura klient/serwer w środowisku Magic'a  
Fig. 9. Client-server architecture in the Magic environment

## 5. Analiza potrzeb Urzędu Miasta w zakresie integracji baz danych

Przedstawiona dalej analiza dotyczy miasta Pszów [3,4]. Prowadzona w tym urzędzie komputeryzacja przebiegała początkowo niezależnie w poszczególnych referatach i wydziałach. Każda jednostka została wyposażona w niezależny system wspomagający pracę tej jednostki. W rezultacie w urzędzie funkcjonuje kilkanaście systemów tworzonych przez różne firmy, w oparciu o różne narzędzia programowe z dominującą rolą Clippera.

Do ważniejszych spośród tych systemów należą:

- system Ewidencja Ludności (Clipper), użytkowany w Referacie Ewidencji Ludności i Spraw Obywatelskich,
- systemy: Ewidencja finansowo-księgowa SFINKS (Clipper), Ewidencja budżetowa SEMP (Clipper), Podatek od środków transportowych POTRANS (Clipper), Ratusz - pakiet



programów ewidencyjnych, wymiaru i księgowości podatkowej (standard Xbase), użytkowane w Wydziale Finansowo-Budżetowym,

- program ewidencji wydawanych koncesji KONCESJA (FoxPro) oraz program ewidencji podmiotów gospodarczych SEDZIG (FoxPro), użytkowane w Referacie Działalności Gospodarczej,
- program ewidencji uprawnień komunikacyjnych KIEROWCA (Clipper) oraz program rejestracji pojazdów REJESTR (Progress), użytkowane w Referacie Komunikacji i Transportu.

Przeprowadzona analiza [3] wykorzystania istniejących w Urzędzie Miasta systemów informatycznych, a także potrzeb w zakresie ich integracji przyniosła m.in. takie wnioski:

- istniejące systemy zostały wszechstronnie sprawdzone i działają bez zarzutu, w związku z czym ważne jest dalsze ich wykorzystanie dla wspomagania pracy urzędu,
- w kilku systemach są przechowywane i użytkowane częściowo te same dane. Prowadzi to do strat czasu przy wprowadzaniu tych danych, a także grozi powstaniem rozbieżności w danych i utratą spójności bazy danych. Dotyczy to głównie danych osobowych mieszkańców. Integracja istniejących systemów mogłaby się sprowadzić w tym przypadku do udostępnienia danych systemu Ewidencji Ludności innym programom użytkowym.

Dla rozwiązania problemu integracji wstępnym krokiem musi być udostępnienie bazy danych systemu Ewidencja Ludności innym programom użytkowym w sieci Urzędu Miasta. Wymóg ten został spełniony przez wprowadzenie zarządzania tą siecią w oparciu o system Windows NT.

W drugim kroku integracji winna nastąpić modyfikacja wszystkich programów użytkowych, w których potrzebne są dane osobowe, dla zapewnienia odczytu tych danych z bazy systemu Ewidencja Ludności. Programy te były jednak tworzone stosunkowo dawno przez różne firmy software'owe. Wprowadzenie wspomnianych poprawek wymagałoby zaangażowania wszystkich tych firm do modyfikacji ich programów. Z przyczyn organizacyjnych zrezygnowano z takiego rozwiązania.

Przyjęte ostatecznie rozwiązanie bazuje na opracowaniu niezależnej, prostej aplikacji<sup>1</sup> udostępniającej do odczytu dane osobowe. Aplikacja ta zostanie zainstalowana we wszystkich systemach użytkowych w Urzędzie Miasta. Po odpowiednim skonfigurowaniu (ustalenie ścieżek dostępu) zapewni to możliwość odczytu danych osobowych systemu Ewidencja Ludności. Niewygodnym problemem pozostaje natomiast fakt, że użytkownicy otrzymają odrębny program, który musi być uruchamiany i obsługiwany niezależnie od podstawowej aplikacji w danym systemie. W tym miejscu z pomocą przychodzi środowisko Windows, do którego są przenoszone wszystkie systemy użytkowe w Urzędzie Miasta. W tym środowisku

---

<sup>1</sup> Program ten został napisany i skompilowany przy użyciu kompilatora Borland Delphi (wersja 1.0); przy czym do komunikacji z bazą danych wykorzystano bibliotekę Apollo firmy Luxent Software dla współpracy z plikami indeksowymi typu NTX.

bowiem, w trakcie pracy podstawowej aplikacji danego systemu, np. rejestracji pojazdów, może zostać uruchomiona w odrębnym oknie nowo opracowana aplikacja zapewniająca dostęp do danych osobowych. W ten sposób dane obu aplikacji będą równocześnie widoczne na ekranie. W szczególnych przypadkach wykorzystanie mechanizmu schowka pozwoli skopiować dane osobowe do bazy danych innej aplikacji.

## 6. Podsumowanie

W pracy dokonano przeglądu mechanizmów i narzędzi integracji baz danych. Przeprowadzono też analizę potrzeb w zakresie integracji zgłaszanych przez użytkowników systemów użytkowych Urzędu Miasta. Analiza ta prowadzi do wniosku, że nieuwzględnienie na etapie projektowania poszczególnych systemów użytkowych przyszłych potrzeb dostępu do lokalnych baz danych musi w przyszłości prowadzić do kosztownych modyfikacji użytkownych programów lub też uzupełniania ich funkcji np. przez tworzenie dodatkowych aplikacji. Zastosowane w przedstawionym przypadku rozwiązanie zaliczyć należy do systemów wielobazowych powiązanych na poziomie języka zapytań.

## LITERATURA

- [1] Bright M.W., Hurson A.R., Pakzad S.: A Taxonomy and Current Issues in Multidatabases Systems. Computer, Vol. 25, No. 3, 1992.
- [2] Dilly R.: Data Mining. An Introduction. Queens University Belfast. Parallel Computer Centre. December 1995.
- [3] Frączek J., Kmonk J., Kozielski S., Pierzchała D.: Przegląd metod i narzędzi integracji baz danych. Projekt Celowy nr 8T11C "System obiegu informacji multimedialnej - pilotujące wdrożenie w Urzędzie Miasta Pszów", 1997.
- [4] Frączek J., Kmonk J., Kozielski S., Pierzchała D.: Koncepcja zmodernizowania systemu zarządzania. Integracja baz danych. Projekt Celowy nr 8T11C "System obiegu informacji multimedialnej - pilotujące wdrożenie w Urzędzie Miasta Pszów", 1997.
- [5] Hammer J. et al.: The Stanford Data Warehousing Project. Stanford University.
- [6] Hurson A.R., Bright M.W., Pakzad S.: Multidatabases Systems: An Advanced Solution for Global Information Sharing. IEEE Computer Society Press, Los Alamitos, California, 1994.
- [7] Inmon W.H.: Building the Data Warehouse. John Wiley & Sons, Inc. 1996.
- [8] Kimball R.: Data Warehouse Toolkit. John Wiley & Sons, Inc. 1996.
- [9] Kleewein J.: Practical Issues with Commercial use of Federated Databases. Proceed.

- of the 22nd VLDB Conference, Bombay, India, 1996.
- [10] Koszłajda T., Morzy T.: Integracja heterogenicznych rozproszonych baz danych z zastosowaniem magazynu danych. Materiały konferencji POLMAN, Poznań 1996.
  - [11] Lorek M.: Analiza istniejącego systemu zarządzania w Urzędzie Miasta Pszów. Projekt Celowy nr 8T11C "System obiegu informacji multimedialnej - pilotujące wdrożenie w Urzędzie Miasta Pszów", 1997.
  - [12] Pierzchała D.: Dostęp aplikacji w SQLWindows do baz danych na różnych serwerach, Zeszyty Naukowe Politechniki Śląskiej seria Informatyka z. 30, 1996.
  - [13] Pierzchała D.: Aplikacja w SQLWindows jako klient bazy danych Ingres, Zeszyty Naukowe Politechniki Śląskiej seria Informatyka z. 31, 1996.
  - [14] Pierzchała D.: Integracja baz danych w heterogenicznych środowiskach sieci komputerowych na przykładzie serwerów MS SQL Server i Gupta SQLBase, Zeszyty Naukowe Politechniki Śląskiej seria Informatyka z. 33, 1997 (w druku).
  - [15] Widom J.: Research Problems in Data Warehousing. Proceed. of 4th Int. Conference of Information and Knowledge Management, Nov. 1995.

Recenzent: Dr hab. inż. Adam Mrózek, Prof. Politechniki Śląskiej

Wpłynęło do Redakcji 5 stycznia 1998 r.

## Abstract

The paper describes the methods and tools of databases integration. It discusses the classification of integration levels and differentiates distributed databases, global schema multidatabases, federated databases, multidatabase language systems, homogenous multidatabase language systems and interoperable systems.

The possibility of databases integration by use of data warehouses is indicated. The solutions based on a data warehouse give flexible foundation for business management systems, decision support systems and specialized systems for advanced and multidimensional data analysis.

The paper introduces several software tools that make a databases integration possible. It pays special attention to the ODBC technology and tools of router/gateway type. It also shows the possibilities of databases integration offered by the MAGIC database application generator.

The last part of the paper contains the analysis of town office information systems integration. The concept of an universal application that assures access to local databases is discussed.