

Bożena MALYSIAK-CIEŚLA, Stanisław CIEŚLA
Politechnika Śląska, Instytut Informatyki

WYKORZYSTANIE TECHNOLOGII ACTIVE SERVER PAGES W DOSTĘPIE DO BAZ DANYCH W SIECIACH INTERNET I INTRANET

Streszczenie. W artykule przedstawiono charakterystykę mechanizmu dostępu do baz danych w sieci Internet/intranet w oparciu o technologię stron aktywnych serwera Active Server Pages i mechanizm uniwersalnego dostępu do danych Universal Data Access.

USAGE ACTIVE SERVER PAGES TECHNOLOGY INTO DATABASE ACCESS THROUGH INTERNET/INTRANET NETWORKS.

Summary. This paper describe characteristic of database access mechanism through Internet/intranet by means of Active Server Pages technology and usage of Universal Data Access technique.

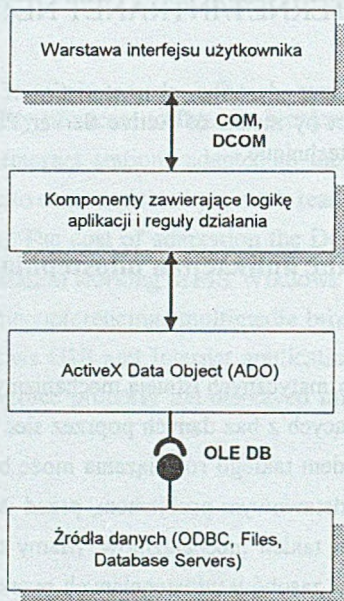
1. Wymagania stawiane aplikacjom udostępniającym dane w sieciach Internet/intranet

W obecnych systemach informatycznych istnieją mechanizmy i technologie pozwalające na udostępnianie danych pochodzących z baz danych poprzez sieć Internet lub intranet szerokiej rzeszy użytkowników. Przykładem takiego rozwiązania może być technologia Internet Database Connector [1], jednak podstawowym problemem, przed jakim stoją twórcy oprogramowania jest poprawa wydajności takich mechanizmów. Mamy tu do czynienia z dużą liczbą użytkowników korzystających z zasobów udostępnianych przez bazy danych i proste rozwiązania oparte na pojedynczym rozszerzeniu serwera WWW, współpracującego ze ściśle określoną bazą danych za pomocą prostego interfejsu, są w tym przypadku niewystarczające.

Kolejnym ważnym aspektem, jaki powinny uwzględniać tworzone mechanizmy, jest zapewnienie skalowalności, pozwalające na łatwe przejście z rozwiązania opracowanego dla niewielkiej sieci lub niewielkiej ilości udostępnianych danych na dużą liczbę odbiorców, lub też ogromne ilości zgromadzonych danych. Postulat skalowalności powinien także uwzględniać możliwość przejścia z prostego systemu bazy danych, opartego na kilku tabelach, na wysoko wydajny serwer lub zastosowanie całkiem nowego systemu gromadzenia i przetwarzania danych, a co za tym idzie architektura taka powinna być otwarta na przyszłe rozwiązania i standardy, jednocześnie akceptując istniejące systemy i różnorodne platformy.

Nowoczesna architektura dostępu do bazy danych powinna także umożliwiać tworzenie aplikacji dla różnych środowisk i w różnych językach programowania, na przykład powinna uwzględniać środowisko sieci lokalnej i jednocześnie te same mechanizmy powinny umożliwiać dostęp do danych poprzez sieć Internet, co spowoduje naturalne rozszerzenie architektury klient-serwer o dodatkową platformę – sieć Internet/intranet. Taka wielowarstwowość architektury pozornie komplikuje cały mechanizm, jednak praktycznie obniża koszty tworzenia konserwacji i utrzymania opracowywanych aplikacji.

Propozycją nowej technologii w dostępie do baz danych uwzględniającą postawione wyżej postulaty jest rozwiązanie firmy Microsoft pod nazwą Uniwersalny Dostęp do Danych (*ang. Universal Data Access*).



Rys. 1. Specyfikacja architektury Universal Data Access

Fig. 1. Universal Data Access Architecture

2. Charakterystyka mechanizmu Universal Data Access

Mechanizm Universal Data Access jest wielowarstwowym modelem pozwalającym na uniwersalny i wysoko wydajny dostęp tworzonym aplikacjom do różnorodnych baz danych zarówno relacyjnych, jak i nierelacyjnych. Specyfikacja modelu UDA została pokazana na rysunku 1. Mechanizm UDA został oparty i opracowany w oparciu o obiektowo zorientowany interfejs programowy COM [2, 3, 4], pozwalając na stosowanie na różnych jego poziomach wielu narzędzi, komponentów i języków programowania.

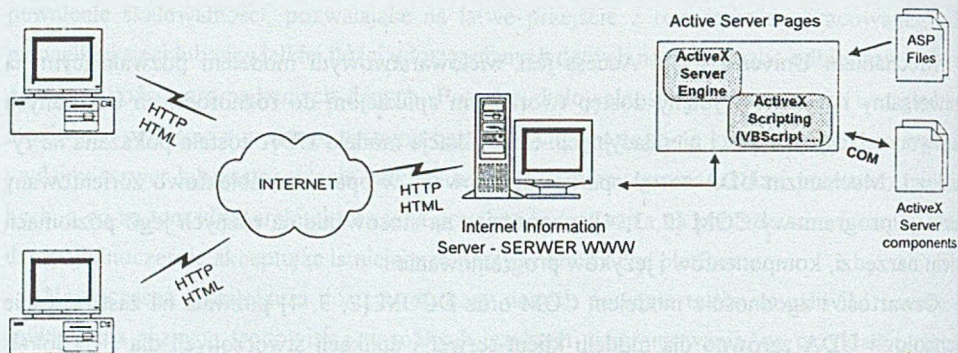
Otwartość i zgodność z modelem COM oraz DCOM [2, 3, 4] pozwala na zastosowanie technologii UDA zarówno dla modelu klient-serwer i aplikacji stworzonych dla sieci lokalnych, jak i wykorzystanie tych samych komponentów dla stworzenia aplikacji pracującej w sieci Internet/intranet. Mechanizm Universal Data Access został wprowadzony jako wykorzystanie technologii Active Server Pages [5], będącej rozszerzeniem serwera WWW firmy Microsoft.

W dalszych rozdziałach zostaną przybliżone poszczególne elementy modelu UDA pozwalające na stworzenie aplikacji udostępniającej dane poprzez sieć Internet/intranet.

2.1. Czym jest technologia Active Server Pages?

Technologia Stron Aktywnych Serwera (*ang. Active Server Pages*) została wprowadzona przez firmę Microsoft jako rozszerzenie kolejnej wersji serwera WWW oferowanego przez tę firmę pod nazwą Internet Information Server [5]. Technologia ASP umożliwia zmianę wyglądu i treści stron WWW na serwerze przed ich przesłaniem do przeglądarki komputera użytkownika w odpowiedzi na jego żądanie dostępu do tej strony. Zmiana treści i wyglądu strony WWW jest możliwa dzięki umieszczeniu w jej treści tekstu programu, w formie skryptu.

Skrypt z instrukcjami, które mają być wykonane, może być umieszczony pomiędzy instrukcjami języka HTML, w miejscu w którym treść strony ma zostać zmieniona i jest zależna od rezultatu wykonania pewnych instrukcji. Jest to technologia podobna do znanych wcześniej rozwiązań pozwalających na umieszczanie w treści strony napisanej w języku HTML fragmentów napisanych w innych językach (np. w językach skryptowych VBScript czy JavaScript), które mogą być wykonane w przeglądarce użytkownika po załadowaniu danej strony. Wadą takiego rozwiązania jest duża zależność tworzonych stron zawierających skrypty od typów przeglądarek, które mają te strony interpretować. Nie wszystkie przeglądarki potrafią wykonywać skrypty, nie wszystkie też wykonują je w ten sam sposób. Taka metoda modyfikowania stron nie jest wygodna, a poza tym treść skryptów jest wprost udostępniona użytkownikom, co w niektórych sytuacjach może być niewygodne, a czasami nawet niebezpieczne.



Rys. 2. Schemat działania technologii Active Server Pages
Fig. 2. Active Server Pages model and technology

Technologia ASP pozwala na wykorzystanie mechanizmu skryptów na stronach WWW, ale pozwala na ich wykonywanie w środowisku serwera WWW i zapewnia, że do przeglądarki na komputerze użytkownika dociera odpowiednio zmodyfikowana już treść strony w języku HTML. Technologia ASP pozwala nie tylko na umieszczanie w treści stron kodu w językach skryptowych, ale także na wywołanie z wnętrza stron WWW zewnętrznych komponentów ActiveX [2, 3], działających w standardzie COM.

Schemat przepływu stron WWW, z użyciem technologii ASP, między przeglądarką użytkownika a serwerem WWW został pokazany na rysunku 2.

Poniżej przedstawiono przykładową treść strony wykorzystującą technologię ASP.

```
<HTML>
<TITLE>Przykład strony aktywnej serwera WWW</TITLE>
<BODY>
<H2>Aktualny dzień miesiąca według kalendarza na serwerze to
<% Call PrintDate %>
</H2>
<SCRIPT LANGUAGE=JScript RUNAT=Server>
    function PrintDate()
    {
        var x
        x = new Date()
        Response.Write(x.getDate())
    }
</SCRIPT>
</BODY>
</HTML>
```

Powyższa strona została napisana z użyciem języka skryptowego JavaScript i powoduje odczyt daty serwera i umieszczenie jej w treści przesyłanej do użytkownika. Poniżej przedstawiono treść tej strony, która zostanie dostarczona do przeglądarki.

```
<HTML>
<TITLE>Przykład strony aktywnej serwera WWW</TITLE>
<BODY>
<H2>Aktualny dzień miesiąca według kalendarza na serwerze to 21
</H2>
</BODY>
</HTML>
```

Zaletą takiego rozwiązania jest ukrycie przed użytkownikami treści skryptów modyfikujących zawartość stron, a równocześnie zapewnienie możliwości dostępu do takich stron za pomocą dowolnej przeglądarki WWW.

2.2. Czym jest OLE DB?

OLE DB to proponowany przez firmę Microsoft systemowy interfejs umożliwiający dostęp do danych. Jest on opracowany w technologii COM i dostarcza niezależnego interfejsu programowego wyższego poziomu, pozwalającego na manipulowanie danymi z poziomu aplikacji niezależnie od ich struktury wewnętrznej.

Komponenty OLE DB mogą być opracowywane dla wszelkich rodzajów systemów gromadzenia i przetwarzania danych, począwszy od prostych tabel, poprzez SQL-owe serwery baz danych do systemów nierelacyjnych. Poprzez opracowanie odpowiednich komponentów OLE DB o jednolitym standardzie interfejsu systemowego szczegóły mechanizmów manipulowania danymi zależne od systemu przechowywania danych są ukryte przed aplikacją i pozwalają na tworzenie modułów aplikacyjnych zawierających jedynie poziom logiki aplikacji i reguły działania niezależne od szczegółów implementacyjnych mechanizmów przechowywanych dane.

Komponenty OLE DB nie muszą się ograniczać jedynie do baz danych, za ich pomocą można przygotować techniki dostępu do systemu zbiorów czy też do informacji przechowywanych niezależnie od bazy danych, na przykład sekwencji wideo.

Takie rozwiązanie pozwala na łatwą migrację gotowej aplikacji między różnymi systemami baz danych. Komponenty zawierające logikę aplikacji zostają zachowane, a zmienia się jedynie komponent i źródło danych. Dokładnie tak samo działa to na poziomie aplikacji, komponenty dostępu do danych i źródła danych zostają te same, ale poprzez wymianę komponentów aplikacji uzyskujemy dostęp do tych samych danych z różnych środowisk i platform systemowych. Na przykład z sieci lokalnej i sieci Internet/intranet.

Mechanizm OLE DB jest mechanizmem niskiego poziomu (poziom systemowy), przez co jest trudniejszy w oprogramowaniu. By ułatwić tworzenie aplikacji wykorzystujących technikę OLE DB, wprowadzono więc dodatkową warstwę pośrednią w postaci uniwersalnych klientów OLE DB w postaci ActiveX Data Objects o prostszym interfejsie programowym.

2.3. Czym jest ADO?

ActiveX Data Object (ADO) jest technologią, która może być użyta przez twórców aplikacji dla sieci Internet/intranet jako uniwersalny interfejs w dostępie do danych. Została ona opracowana jako aplikacyjny interfejs do mechanizmu OLE DB, który to można potraktować jako mechanizm niskiego poziomu dostępu do danych. ADO miała być technologią ułatwiającą tworzenie aplikacji wykorzystujących mechanizmy OLE DB. Technologia ADO została opracowana także, by zmniejszyć obciążenia sieci wynikające z transferu danych i poprawić wydajność tworzonych aplikacji.

Dla obiektów ADO nie ma znaczenia, jaki system odpowiada fizycznie za przechowywanie danych, jest to tylko warstwa pośrednicząca w operacjach na danych, co powoduje, że można jej używać bez modyfikacji po wymianie systemu przechowującego dane.

Technologii ADO można używać z poziomu wielu języków programowania, można tworzyć aplikacje w C++, Visual Basic'u czy też w Java'ie. Można także wprost odwoływać się do metod i obiektów związanych z ADO z treści skryptów napisanych w językach skryptowych technologii stron aktywnych ASP. Co otwiera drogę twórcom aplikacji przeznaczonych dla sieci Internet/intranet do świata różnorodnych systemów baz danych.

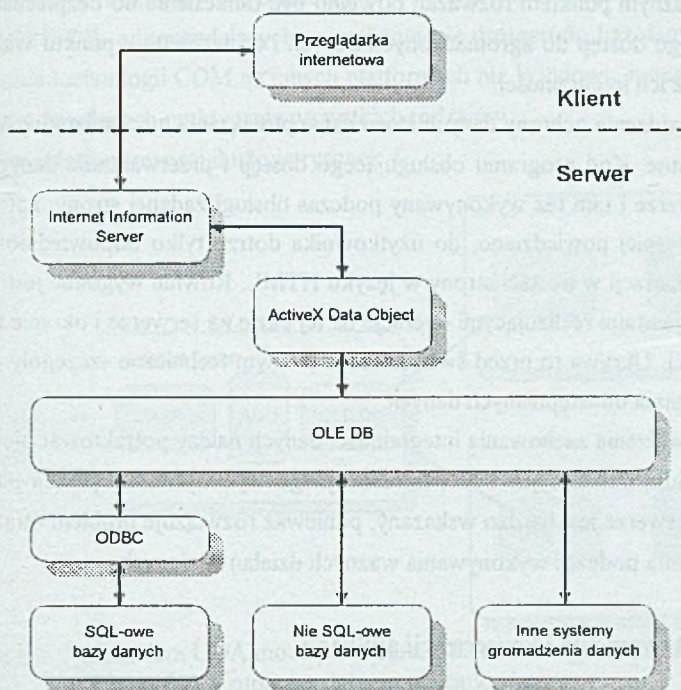
Podstawowymi obiektami technologii ADO są obiekty: Connection, Command i Recordset, oraz zbiór kilku podobieństw reprezentujących dane, metody dostępu, właściwości i błędy, jakie mogą pojawić się podczas operacji na danych. To ten prosty zbiór obiektów stanowi cały interfejs mechanizmu ADO. Sam mechanizm jest już zintegrowany i dostępny na platformach Windows 95 i Windows NT, a ma stanowić podstawowy element przyszłych wersji tych systemów operacyjnych.

3. Jak model Universal Data Access oparty na technologii Active Server Pages może być stosowany dla sieci Internet/intranet

Technologię Universal Data Access można wprost zastosować dla aplikacji Internetowych w oparciu o serwer WWW firmy Microsoft Internet Information Server od wersji 3.0 w górę. Jak już wspomniano we wcześniejszych rozdziałach, komponent ActiveX Data Object jest oparty na technice ActiveX [2, 3, 4], w związku z czym może zostać użyty wprost na stronie WWW wykonywanej na serwerze zapisanej jako strona Active Server Page.

Metody dostępu do danych realizowane poprzez metody obiektu ADO, wywoływane z treści skryptu zawartego na stronie WWW powodują wywołanie odpowiednich komponentów OLE DB zawartych w serwerze Internetowym oraz w systemie operacyjnym, a te z kolei

realizują bezpośrednie operacje na danych. Schemat takiego rozwiązania przedstawia rysunek 3.



Rys. 3. Jak OLE DB i ADO współpracują z serwerem Internetowym
Fig. 3. How OLE DB and ADO work over Internet server

3.1. Wydajność mechanizmów UDA i ASP w aplikacjach Internetowych

Połączenie mechanizmu UDA i ASP na serwerze Internetowym jest wydajne z punktu widzenia wykonywanej aplikacji. Całość operacji związanych z logiką aplikacji, regułami decyzyjnymi i operacjami na danych jest wykonywana na serwerze, a użytkownik za pomocą komputera i przeglądarki wydaje jedynie polecenia i obserwuje otrzymane rezultaty.

Wydajność serwera można poprawić także nie tylko wprowadzając bardziej wydajne serwery baz danych, ale także wprowadzając mechanizm pośredni, jakim jest serwer Microsoft Transaction Server [3, 4], którego zadaniem jest zoptymalizowanie wykonywania komponentów przy wielodostępie do umieszczonych w nim obiektów OLE DB. Może to w znacznym stopniu ułatwić tworzenie aplikacji, a także spowodować, iż podniesie się komfort pracy użytkowników.

Takie rozwiązanie spełnia postulat skalowalności, w przypadku niewystarczającej wydajności systemu nie trzeba wprowadzać nowego systemu, a jedynie wymienić te komponenty, które stanowią wąskie gardło aplikacji.

3.2. Aspekty bezpieczeństwa w dostępie do danych

Bardzo ważnym punktem rozważań powinno być odniesienie do bezpieczeństwa systemu umożliwiającego dostęp do zgromadzonych danych i to zarówno z punktu widzenia ochrony danych, jak też ich integralności.

Z punktu widzenia ochrony danych stosowanie powiązania mechanizmów ASP i UDA jest bardzo korzystne. Kod programu obsługującego dostęp i przetwarzanie danych jest umieszczony na serwerze i tam też wykonywany podczas obsługi żądanej strony, natomiast zgodnie z tym, co wcześniej powiedziano, do użytkownika dotrze tylko odpowiednio spreparowany rezultat tych operacji w postaci strony w języku HTML. Równie wygodne jest połączenie bazy danych z obiektami realizującymi operacje na tej bazie na serwerze i ukrycie tego wewnątrz chronionej sieci. Ukrywa to przed światem zewnętrznym techniczne szczegóły przechowywania i przetwarzania udostępnianych danych.

Z punktu widzenia zachowania integralności danych należy potraktować sieć Internet jako zawodne medium transmisyjne i fakt transakcyjnego wykonywania operacji na bazie danych w całości na serwerze jest bardzo wskazany, ponieważ rozwiązuje problem utraty komunikacji z aplikacją klienta podczas wykonywania ważnych działań na danych.

3.3. Koszt tworzenia i konserwacji aplikacji

Wielowarstwowość i otwartość prezentowanej architektury pozwala na wykorzystywanie istniejących elementów do tworzenia nowych i modyfikowania istniejących rozwiązań bez konieczności tworzenia systemu od początku. Zastosowanie technologii COM i DCOM w omawianym modelu pozwala na opracowanie aplikacji dla różnych środowisk w oparciu o dokładnie te same moduły logiki aplikacji i moduły dostępu do danych różniące się jedynie interfejsem klienta. W ten sposób jednocześnie można przygotować wersje aplikacji dla sieci lokalnej i dla sieci Internet/intranet.

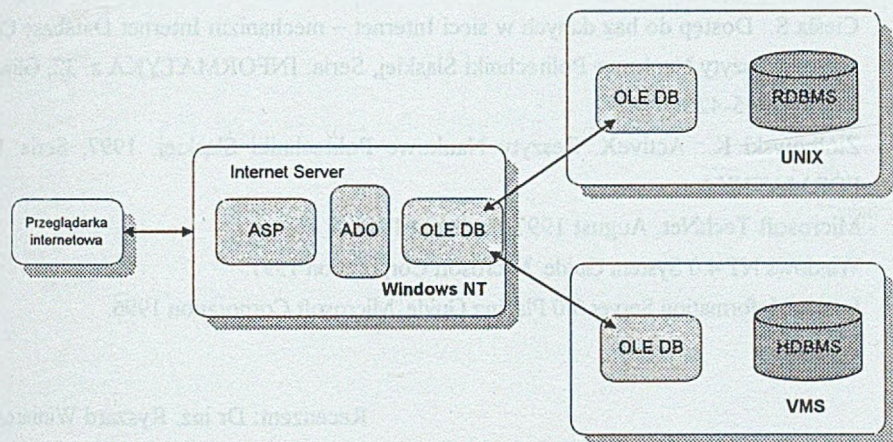
Podobnie wygląda sytuacja z dostępem do danych, dzięki komponentom OLE DB można dla jednej aplikacji i jej modułów logiki oraz reguł decyzyjnych wymieniać komponenty operujące na danych, co wprowadza dużą elastyczność w tworzeniu aplikacji. W przypadku konieczności migracji danych do innych systemów obniża koszty modernizacji aplikacji.

Fakt stosowania jednolitego interfejsu w dostępie do danych pozwala na tworzenie aplikacji operujących na różnorodnych źródłach danych, co powoduje proste tworzenie interfejsu dla heterogenicznych systemów gromadzenia danych. Unika się w ten sposób konieczności migracji już istniejących danych do nowych systemów, by udostępnić je w innych środowiskach.

3.4. Wieloplatformowość w dostępie do danych

Dzięki stosowaniu wcześniej wspomnianego otwartego interfejsu COM w tworzeniu komponentów OLE DB, odpowiadających za realizowanie dostępu do baz danych oraz możliwości stosowania technologii COM na innych platformach niż Windows, można uzyskać dostęp do systemów baz danych praktycznie wszystkich rodzajów.

Przykład wieloplatformowości ilustruje rysunek 4.



Rys. 4. Mechanizm UDA może przetwarzać dane z różnych platform

Fig. 4. UDA supports data on multiple platforms

4. Podsumowanie

Artykuł ten nie miał na celu przedstawienia rozwiązania najlepszego, które będzie stanowił przełomowy krok w tworzeniu aplikacji dla sieci Internet/intranet, lecz zasygnalizowanie pewnych trendów rozwojowych i kierunków, w jakich rozwija się idea aplikacji o charakterze bazodanowym dla sieci Internet/intranet.

Na obiektywne oceny praktyczne przedstawianego rozwiązania ze strony autorów artykułu przyjdzie jeszcze poczekać, gdyż omawiane zagadnienie jest rzeczą nową i wymaga pewnych nakładów prac koniecznych do opracowania przykładowych aplikacji, takich jak opracowane dla mechanizmu Internet Database Connector [2], którym autorzy zajmowali się wcześniej. Jednak już na przykładzie analizy architektury i wykonanych prostych badań wynika, iż w stosunku do swoich poprzedników zauważa się duży postęp w technologii. W proponowanym mechanizmie wyeliminowano wiele ograniczeń, wprowadzono otwartość standardów, poprawiono aspekty bezpieczeństwa i zwiększono wydajność.

Można stwierdzić, że technologia Active Server Pages w połączeniu z modelem Universal Data Access stanowi atrakcyjną platformę do prac nad tworzeniem zaawansowanych aplikacji bazodanowych w sieciach Internet i intranet.

LITERATURA

- [1] Cieśla S.: Dostęp do baz danych w sieci Internet – mechanizm Internet Database Connector. Zeszyty Naukowe Politechniki Śląskiej, Seria: INFORMATYKA z. 32, Gliwice 1997, s. 415-426.
- [2] Ziółkowski K.: ActiveX. Zeszyty Naukowe Politechniki Śląskiej, 1997, Seria: INFORMATYKA
- [3] Microsoft TechNet. August 1997. Redmont 1997.
- [4] Windows NT 4.0 System Guide. Microsoft Corporation 1997.
- [5] Internet Information Server 3.0 Planing Guide. Microsoft Corporation 1996.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 22 grudnia 1997 r.

Abstract

This article discusses access to database through Internet using mechanism Universal Data Access and Active Server Pages technology. The article is divided into four parts.

The first one discusses problems of high performance access to data from Internet's point of view and scalability for this purpose.

The second part describes characteristic of Universal Data Access technology and possibility of using these methods of Internet server. Interface between Active Server Pages as Internet server and database by means of Universal Data Access (OLE DB and ActiveX Data Objects) connectivity is also described.

The third section discusses an idea of how Universal Data Access works on Active Server Pages. In this part high performance of data access and multiple platform support are explained. Also aspects of security and creating application costs are discussed.

The last parts shows short comparison of Universal Data Access, described in this paper, and Internet Database Connector technology. Also future benefits of UDA and ASP connection are shown.

REFERENCES

1. Krawczyk – W., *Nowoczesne Inżynierie Sztucznej Inteligencji i Systemów*

TECHNOLOGIA REALIZACJA REKURENCYJNYCH ALGORYTMÓW LOGICZNO-ALGEBRAICZNYCH

Streszczenie. W artykule przedstawiono algorytm równoległej realizacji rekurencyjnych procedur algebraicznych dla zadania rozpoznawania faktów w systemach ekspertycznych. Wykorzystano technikę rekurencyjnych procedur wyboru i dekompozycji faktów. Zaprezentowano sposób obliczeń w systemie wieloprocesorowym i za pomocą algorytmu równoległego realizowały obciążenie procesorów. Przedstawiono wyniki badań pomiarów obciążeniowych dla wybranej tabeli.

PARALLEL EXECUTION OF RECURSIVE LOGIC-ALGEBRAIC ALGORITHMS

Summary. The paper presents algorithm of parallel computation for the knowledge-based method applied to a class of knowledge-based systems. The task of recursive procedure based on the decomposition of facts is presented. The parallel computation approach is suggested and dynamic load balancing algorithm is proposed. The results of computational experiments are also presented.

W artykule opisano równoległą realizację rekurencyjnych procedur algebraicznych dla zadania rozpoznawania faktów w systemach ekspertycznych. Wykorzystano technikę rekurencyjnych procedur wyboru i dekompozycji faktów. Zaprezentowano sposób obliczeń w systemie wieloprocesorowym i za pomocą algorytmu równoległego realizowały obciążenie procesorów. Przedstawiono wyniki badań pomiarów obciążeniowych dla wybranej tabeli.