UNIVERZITET U NIŠU
EKONOMSKI FAKULTET
Časopis "**EKONOMSKE TEME**"
Godina izlaženja XLVII, br. 2, 2009., str. 113-126
Adresa: Trg kralja Aleksandra Ujedinitelja 11, 18000 Niš
Tel: +381 18 528 601 Fax: +381 18 523 268

SISTEM PODRŠKE ZA MARKETING ODLUČIVANJE

Prof. dr Momčilo Đorđević*
Doc. dr Željko Marčićević•

Rezime: *Sistem podrške za marketing odlučivanje uključuje koordiniran skup podataka, sistema, sredastava i tehnika, sa podržavajućim softverom i hardverom u cilju prikupljanja, analize, interpretacije i prezentacije informacija donosiocima marketing odluka. Tako se olakšava rukovodiocima marketinaga bolje razumevanje a time i rešavanje određenog problema.U radu se razmatra koncept, struktura i arhitektura višeslojnih baza podataka radi što bolje analize i implementacije modela informacionog sistema. Nova višeslojna bazna arhitektura se nameće kao osnova za realizaciju distribuiranih baza podataka na deljenom prostoru Interneta.*

Ključne reči: *Baza podataka, Athitektura, Internet.*

1. Uvod

Podatak je neka kodirana činjenica iz realnog sistema, on je nosilac informacije. Informacija je protumačeni (interpretirani) podatak. Interpretacija podataka se vrši na osnovu strukture podataka, semantičkih ograničenja na njihove vrednosti i preko operacija koje se nad njima mogu izvršiti [4].

Baza podataka (BP) je kolekcija medjusobno povezanih podataka, uskladištenih sa minimumom redundanse, koje koriste svi procesi obrade u sistemu. Podaci su jedinstveni resurs u nekom sistemu i njima se mora upravljati na jedinstven način, onako kako se upravlja i drugim vitalnim resursima poslovnih sistema.

* Ekonomski fakultet Kragujevac
• Visoka poslovna škola strukovnih studija u Novom Sadu.
UDK 658.8:004.738.5; Pregledni rad
Primljeno: 28.04.2009.

Osnovu informacionog sistema čini baza podataka, koja se sada može definisati i kao kolekcija međusobno povezanih podataka, koja modelira (prikazuje) objekte, veze objekata i atribute objekata posmatranog realnog sistema. Ona zbog toga predstavlja fundamentalne, stabilne, sporo izmenljive karakteristike sistema, objekte u sistemu i njihove međusobne veze. Zato se projekat IS mora bazirati na bazi podataka. Ako je baza podataka dobar model stanja realnog sistema, ako programi za održavanje dobro modeliraju dejstvo ulaza na stanje realnog sistema, onda će se bilo koja informacija potrebna za upravljanje (izlazi), čak i one unapred nepredvidjene, moći dobiti iz IS.

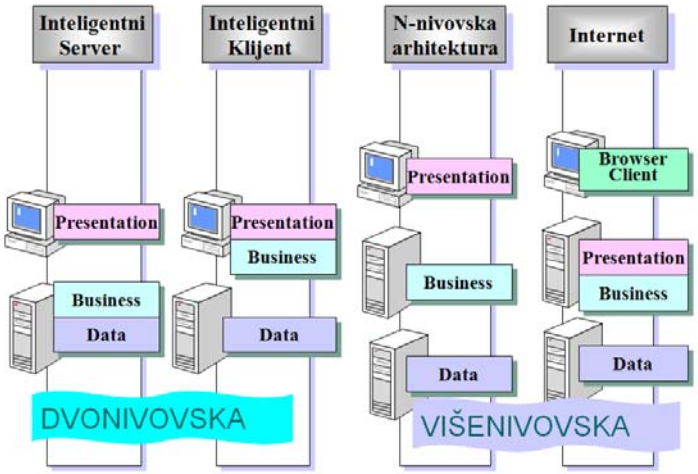
2. Vrste baznih arhitektura

Hronološki gledano (slika 1.) razlikujemo [4]:

- Jednoslojnu arhitekturu
- Dvoslojnu arhitekturu
- Troslojnu arhitekturu
- Višeslojnu arhitekturu

U troslojnom generičkom modelu jasno se odvaja upravljanje podacima, aplikaciona logika i korisnički interfejs. Prilagodljivost brzim promenema, kako u korisničkom (poslovnom), tako i u implementacionom (tehnološkom) okruženju. Suštinu ove arhitekture odražava srednji sloj koji se različito naziva: aplikacioni server, transakcioni server, server komponenti, server poslovnih pravila, čime se posebno ističe neka funkcionalnost ovoga sloja. Jasno se odvaja upravljanje podacima, aplikaciona logika i korisnički interfejs. Prilagodljivost brzim promenema, u poslovnom i implementacionom okruženju omogućava i transparentno povezivanje korisničkih aplikacija sa različitim izvorima podaka na raznim platformama, a ne samo sa jednim serverom baze podataka.

Koncept distribuiranih softverskih komponenti (CORBA, DCOM, Java Beans) omogućava da se i komponente srednjeg sloja distribuiraju. Troslojna arhitektura je generička za višeslojne arhitekture koje postaju opšteprihvaćeni standard. U njima se različite funkcije srednjeg sloja (“middleware”) raslojavaju, da bi se preko većeg broja slojeva, odnosno većeg stepena indirekcije, omogućila veća modularnost, heterogenost i elastičnost sistema



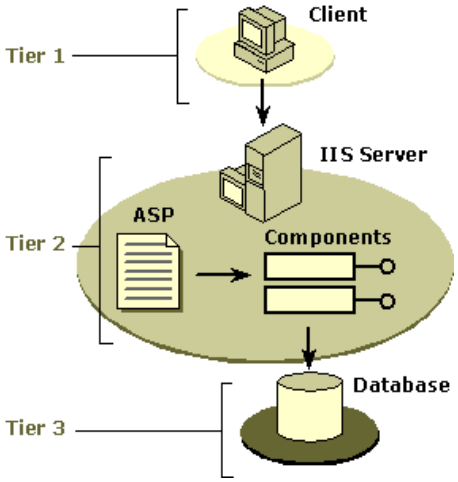
Slika 1. Verzije arhitektura

3. Distribuirano bazno okruženje

Današnje klijent/server aplikacije liče malo na svoje pretke da im je dato novo ime, multitier (mnogo-nizovna) aplikacija, takođe poznata kao n-tier arhitektura. U ovom modelu proces se dešava između klijenta i servera, a logika posla se nalazi u srednjem nizu. Većina sistema će izvoditi tri glavna zadatka, koji se podudaraju sa tri niza ili slojeva u n-tier modelu:

- Označeni niz 1 u priloženoj slici 2. obuhvata svo korisničko iskustvo. Ne samo da ovaj sloj obezbeđuje grafički interfejs tako da se korisnici mogu povezati sa aplikacijom, ubaciti podatak, pogledati rezultate koji su zahtevani, već takođe omogućava manipulisanje i formatiranje podataka jednom kada ih klijent dobije. U Web aplikacijama, browser prikazije podatke ovog sloja [2].

- Niz 2, između interface i podataka servisnih slojeva, je glavni od distribuiranih aplikacionih razvoja. Poslovna logika, koja sadrži pravila koja upravljaju procesnim aplikacijama, povezuje korisnika na jednom kraju sa podatkom na drugom. Funkcije kojima upravljaju pravila izbliza oponašaju svakodnevne poslovne podatke, i mogu biti samo jedan zadatak ili više zadataka.



Slika 2. ASP bazna distributivna arhitektura

• Pokazano kao niz 3 na prikazanoj slici 2, servisi podataka su obezbeđeni struktuiranim (SQL, Oracle bazom podataka) ili nesrukturiranim (Microsoft Excange, Microsoft Message Queuing) skladištima podataka, koja upravlja i obezbeđuje pristup aplikacionom podatku. Sama aplikacija može osigurati servise jednog ili više skladišta podataka [2].

Arhitektura sa tri niza izoluje svaki značajan deo funkcionalnosti, tako da je prezentacija nezavisno od procesnih pravila i poslovne logike, koji je u povratnom odvojen od podatka. Ovaj model zahteva mnogo više analiziranja i spoljašnjeg dizajniranja, ali najveće smanjenje je i dalje cena i povećanja funkcionalne fleksibilnosti u dužem periodu.

Struktura i arhitektura višeslojnih baza podataka su deo sistema podrške marketing odlučivanja (SPMO) [8]:

Slika 3. Koncept SPMO – Sistem podrške marketing odlučivanja

Microsoft je razvio Windows distribuiranu aplikacionu arhitekturu (Windows DNA) kao put koji će potpuno integrisati Web sa n-tier modelom razvoja. Windows DNA definiše uređenje za solucije isporuke koje sreću visoke zahteve korporativnog kompjuterisanja, Internet-a, i globalnu elektronsku trgovinu, dok se smanjuju krajnji troškovi razvoja i cene plata.

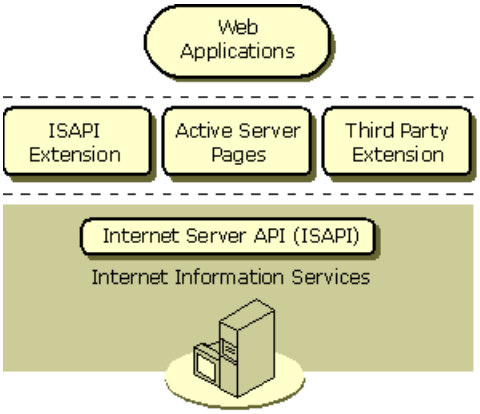
Windows DNA arhitektura upotrebljava standardne servise bazirane na Windows-u da bi odredila potrebe za svaki niz u višenizovnoj soluciji: koristeći interface i navigaciju, poslovnu logiku i skladišta podataka. Servisi korišćeni u Windows DNA –u, koji su integrisani kroz Component Object Model (COM) sadrže:

- Dynamic HTML (DHTML)
- Active Server Pages (ASP)
- COM components
- Component Services
- Active Directory Services
- Windows security services
- Microsoft Message Queuing
- Microsoft Data Access Components

IIS je sastavni deo Windows DNA arhitekture [1]. Važna uloga IIS-a je da povezuje klijente pristupajući sistemu kroz Hipertext Transfer Protocol (HTTP) sa ostalim Windows DNA servisima, kao što su DHTML, ASP, itd. IIS sadrži osnovni sistem funkcionalnosti koji sistemi razvoja mogu pružiti da bi definisali običnu aplikacionu arhitekturu slika 3.

Ovaj deo sadrži:

- **IIS Core Funcionality:** Opisuje osnovne funkcionalnosti koje možete da koristite da bi napravili Web aplikaciju
- **IIS and Component Services:** U osnovi se funkcionalnost koristi da izoluje, upravlja i koordiniše procesiranje za obavljanje ASP aplikacija
- **IIS Request Processing:** Opisuje procenu procedure upotrebljenu da odredi tip zahteva



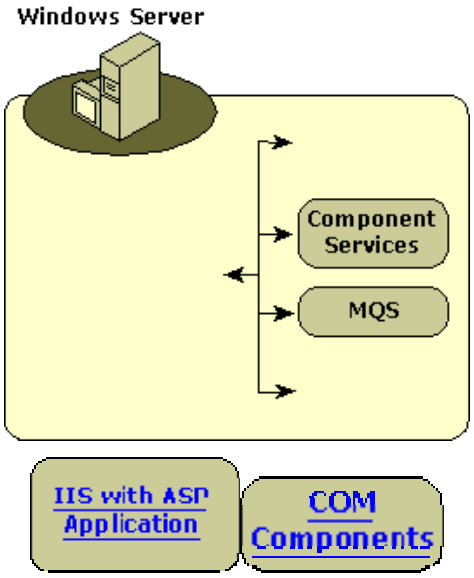
Slika 4. Internet Information Services - IIS

ASP produžuje ovu funkcionalnost obezbeđujući vezu sa COM arhitekturom i prema tome i sa ostalim učesnicima u Windows DNA [1]. Takođe, može se produžiti IIS arhitektura definisanjem običnog seta funkcija koristeći ISAPI. Veza između IIS bitnih funkcionalnosti, ASP, i produživanje arhitektura je prikazana na slici 3.

IIS i servisi komponenata sarađuju zajedno da bi oformili osnovu arhitekture za pravljenej Web aplikacija. IIS koristi funkcionalnost obezbeđenu servisima komponenata da:

- Izoluje aplikaciju u jasne procese
- Upravlja komunikacijom između COM komponenata (uključujući ASP stvorene objekte)
- Koordinira transakcije procesirane za prenosive ASP aplikacije

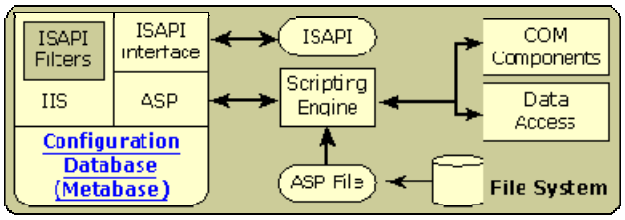
Objekti u ovom konceptu opisuju osnovne funkcionalnosti i opisuju veze IIS arhitekture sa ostatkom Windows DNA arhtekture dato na slici 4. IIS definiše osnovne funkcionalnosti koje se mogu koristiti da bi se napravia Web aplikacija. Active Server Pages (ASP) i druge Microsoft tehnologije pružile su osnovne funkcionalnosti da bi stvorile bogato okruženje za razvoj aplikacije [1].



Slika 5. Veza IIS sa ostatkom aplikacija

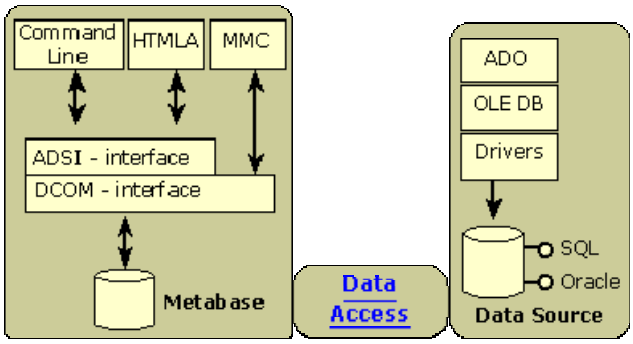
Osnovna funkcionalnost servera je izložena kroz Internet Server Application Programmer Interface (ISAPI) dato na slici 5. Bitne funkcije koje IIS obezbeđuje sadrže [2]:

- Utvrđivanje i očuvanje HTTP veza
- Čitanje HTTP zahteva i pisanje HTTP odgovora
- Modifikovanje HTTP zaglavlja
- Obezbeđivanje klijentu informacije o sertifikatu
- Upravljanje asinhronih veza
- Mapping Uniform Resource Locators (URLs) na fizičke delove
- Upravljanje i pokretanje aplikacija
- Prenošenje podataka



Slika 6. ISAPI

U IIS verziji 4.0, Microsoft Transaction Server (MTS) ibezbeđuje transakcionu podršku. U IIS 5.0 i Windows 2000 servisi komponenata omogućuju sve transakcije koje podržava MTS, u povećanju broja drugih komponenata razvojnih karakteristika. U IIS verziji 4.0 transakciona podrška je obezbeđena od strane Microsoft Transaction Server (MTS). Za IIS 5.0 i Windows 2000 Servisi komponenata obezbeđuju svu podršku transakcijama MTS-a, sa dodatkom brojeva ostalih osobina. IIS definiše Web aplikacije kao grupu izvornih podataka koji su grupisani u logički namespace. Grupisanjem izvora u aplikacije stiče se sposobnost da se poseduju podaci duž namespace i da se pokreću aplikacije u izolovanom procesu. U unutrašnjosti IIS koordiniše izolovane aplikacije kroz objekat poznat kao Web Application Manager. Ovaj objekat sadrži javni interface (IWAMAdmin) koje možemo da koristimo da bi se napravili programi za administriranje Web aplikacija. Kada se pokrene Web aplikacija u izolovanom procesu, IIS koristi servise komponenata da koordiniše zajedničkim pristupom izvorima i prosledi dodatak informacija između COM komponenata dato na slici 6.



Slika 7. COM komponente, ADO i OLE DB Interface

IIS koristi servise komponenata ObjectContext. Na primer, ako pravimo COM komponentu u Visual Basic-u u kojoj je potrebno da pristupi potčinjenom obliku iz HTML fajla, možemo koristiti dati kod [5]:

```
Dim objObjectContext As ObjectContext
Dim vntIn As Variant
Set objObjectContext = GetObjectContext ()
vntIn = objObjectContext.Item("Request").Form("Field1")
```

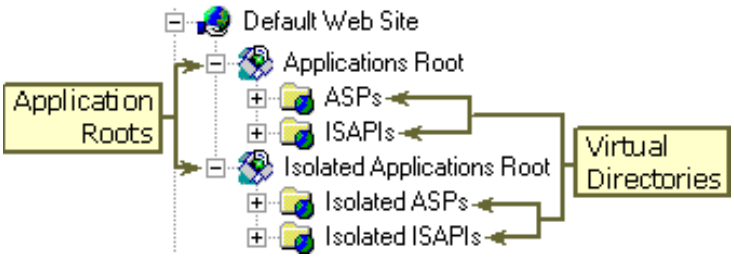
Postoji bogato i kontinualno širenje seta alata za razvoj Web aplikacija, osnovni krug razvoja je sličan sa razvojem Desktop aplikacija u mnogim pogledima [6]:

- **Defining Application Boundaries:** Opisuje neophodne korake potreben da se montira vaše ASP stranica u jednu Web aplikaciju.
- **Controlling Application Flow:** Skiciranje ishoda da bi se kontrolisao tok Web aplikacija.
- **Accommodating International Clients:** Sadrži informaciju koja ima veze sa lokalizacijom Web sajta u raznim jezicima.

ASP bazirana aplikacija je kolekcija ASP strana i COM komponenata. Kada se definiše aplikacija, koriste se interni informacioni servisi koji su tu da odrede aplikacijsku startnu tačku direktorijuma u Web sajtu. Svaki fajl i folder koji se nalaze pod startnom tačkom direktorijuma u Web sajtu je značajan deo aplikacija. Prema tome može se koristiti struktura direktorijuma da bi se formisrala application boundaries koji definišu područje aplikacije. Može postojati više od jedne aplikacije po sajtu i svaka aplikacija može biti drukačije oblikovana.

Jedan od najvažnijih zadataka je okupljanje ASP stranice u jednu Web aplikaciju. IIS koristi koncept namespace da identifikujete aplikacije.

Namespace je put asocijacija u području memorija sa lako prepoznatljivim imenom; on identifikuje grupu fajlova kao zajedničku svojinu. IIS koristi virtualne direktorijume da definiše namespaces za aplikacije. Sledeća slika 7 ilustruje ovaj koncept:



Slika 8. Virtuelni direktorijumi generisani u IIS-u

Skripte i ISAPI proširenja DLLs unutar granica aplikacije formira izolacionu jedinicu koja je uvek pokreće u procesu sa jednim serverom IIS administratori mogu čak pokretati aplikacije u istom server procesu kao i IIS, izmenjeni proces (proces sa nedostacima), ili mogu izolovati aplikaciju pokretanjem izolovanog procesa koji je delimično koristan tokom razvoja kao i testiranja [6].

Pet mogućih konfiguracija za aplikacije su:

- Staviti sve .asp fajlove i komponente u isti proces kao IIS; Ovakva konfiguracija obezbeđuje najbrže performanse
- Staviti sve .asp fajlove i komponente u jedan proces i IIS u drugi proces
- Staviti sve .asp fajlove i komponente u izmenjenom procesu, IIS u drugi proces i specijalne aplikacije u izolovani proces
- Staviti sve .asp fajlove i IIS u jedan proces i komponente u drugi proces
- Staviti sve .asp fajlove u jedan proces, komponente u drugi i IIS u treći proces

U ranijim verzijama IIS-a sve ISAPI aplikacije (uključujući ASP) delile su resurse i memoriju server procesa. I pored ovih obezbeđenih brzih performansi nestabilne komponente mogu prouzrokovati da server "padne".

Ako se pokreće aplikacija kao odvojen proces, ili sa drugim aplikacijama u jednom izmenjenom procesu, mora se odabrati High (Isolation) ili Medium (Pooled) iz Application Protection padajuće liste Home Directory ili Virtual Directory sopstvenom listu. Prvo treba da se kreira aplikacija za aplikacionu startnu tačku direktorijuma. Komponente

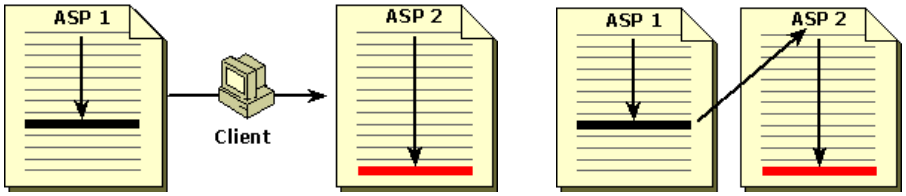
koje će biti pokrenute u novom procesu moraju biti instalirane u odgovarajućoj COM aplikaciji.

Aplikacije isključene iz procesa i komponete, uključujući ISAPI produženja, nisu u mogućnosti da pristupe metabazičnim osobinama. Ova restrikcija je primarna iz sigurnosnih razloga da spreče ne autorizovane promene na meta bazi. Ako želitmio da dopustimo aplikacijama koje su isključene iz procesa da pristupe meta bazi moramo učiniti sledeće:

- Dati IWAM_machinename izveštajni pristup meta bazi
- Promeniti identitet COM aplikacija isključenih iz procesa od interaktivnog korisnika (IUSR_machinename) specifičnom korisniku računa i dati tom računu pristup meta bazi

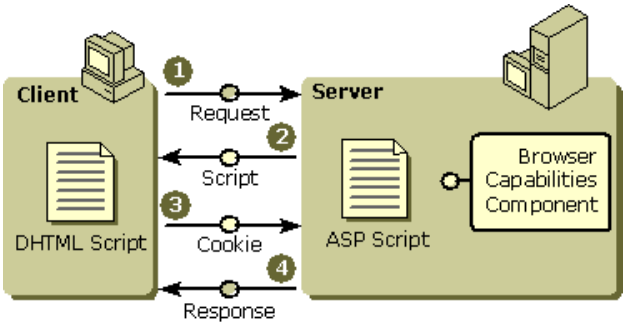
ASP obezbeđuje šest različitih puteva uticanja na glavni tok izvršenja. Ovih šest metoda su prikazani u sledećem dijagramu slika 8. Strelice prikazuju tok izvršenja [4]:

- Redirekcija
- Transferring
- Executing
- Component invocation
- Exiting
- Procedural procesing



Slika 9. Redirekcija i transferring

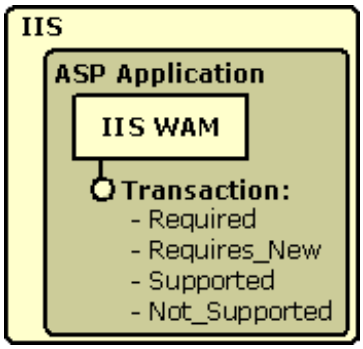
U IIS 5.0 Browser Capabilities component su napredovale da premaše ranije limite dizajna to može biti modifikovano za individualne zahteve dok klijent vraća cookie opisujući njegove sposobnosti. Ako inicirani zahtev. asp fajla ne sadrži cookie možete opozvati deo skripte koje će pšokrenuti ka klijentu da stvori cookie sledeća instrukcija pikazuje sekvence lokacije slika 9.



Slika 10. Komunikacija klijent server

4. Tehnologija transakcione obrade

Osnovna tehnologija koja omogućava Active Server Pages (ASP) da učestvuju u transakcijama je Component Services, koji pruža IIS-u transakcione servise kao i okruženje za instance hosting komponenti. Jedna od prednosti ovog okruženja je mogućnost kreiranja atributa za instance pojedinih komponenti. Kada IIS kompajlira skript u ASP stranici, kreira se nova instanca IIS Web Application Manager-a (IISWAM) slika 10.



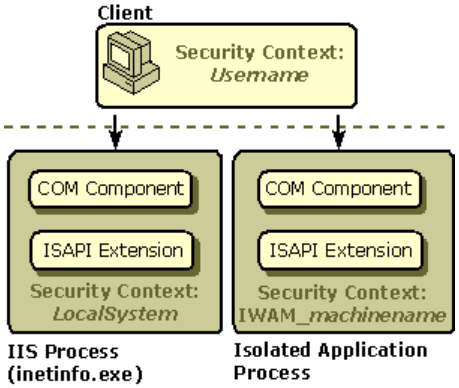
Slika 11. Tehnologija transakcije

IIS WAM je Component Object Model (COM) komponente koju IIS koristi za nadgledanje aplikacija. Ako skript sadrži TRANSACTION = Required u svoj skript, rekli smo Component Services-u da instanca IISWAM-a, koju on kreira, treba da izvrši transakciju. Ako skript ne ASP stranici kreira instancu bilo koje druge komponente koja je registrovana sa Component Services, Component Services će je tretirati kao deo iste transakcije. Sledeći dijagram ilustruje vezu ASP-a i IISWAM-a. Component

Services pružaju IIS-u transakcioni servis preko dva različita sloja. Na najnižem sloju, Component Services sarađuje sa Microsoft Distributed Transaction Coodinator-om (MS DTC) da bi garantovao da će transakcije udovoljiti ACID zahtevima (atomski/nedeljiv, konzistentan, izolovan, postojan) pouzdanog sistema sa transakcionom obradom. Component Services povezuje instance komponenti sa MSDTC-om preko dva mehanizma: menadžera resursa i dispanzera resursa.

5. Korišćenje sigurnosnog konteksta

Windows uspostavlja sigurnosni kontekst za svakog korisnika koji je ulogovan. Kada IIS primi zahtev od klijenta on autorizuje zahtev i onda prepoznaje klijenta. Dok IIS prepoznaje klijenta, IIS deluje unutar oblasti sigurnosnog konteksta autorizovanog korisnika. Sigurnosni kontekst može biti promenjen tokom raznih etapa obrade zahteva, zavisno od prirode zahteva klijenta. Sledeća slika 11 ilustruje razne sigurnosne kontekste koji mogu da učestvuju u obradi zahteva.



Slika 12. ISAPI

Sigurnosni kontekst IIS procesa (inetinfo) je poznat kao Local System. Međutim kada IIS obrađuje zahtev klijent, on prepoznaje kontekst klijenta koji je generisao zahtev. Ako je klijent autorizovan sa Anonymous autorizacionom šemom, sigurnosni kontekst će biti IUSR_machinename za aplikacije u procesu i IWAM_machinename za aplikacije koje se izvršavanju u izolovanom procesu. Ako je klijent autorizovan sa bilo kojom drugom autorizacionom šemom, sigurnosni kontekst će se mapirati/preslikati na nalog klijenta. Mnoge IIS aplikacije zahtevaju resurse koje pružaju druge

softverske komponente. Na primer, DLL proširenje ISAPI-ju može da pozove Automation Server softverske kompanije kao trećeg lica, ili CGI aplikacija može da startuje program razvijen na Microsoft Visual Basic-u. Ove komponente zahtevaju, trajne informacije čuvane u registry koje mogu da traže radi korektnog izvršavanja.

6. Zaključak

Distribuirani IS daje odgovor na zahteve savremenog poslovanja. Distribuirani IS omogućuje [5]:

- Pristup svim relevantnim strukturama podataka
- Prezentaciju konkretnih sintetičkih informacija
- Donošenje odluke uz saznanje o uzrocima i posledicama
- Trenutno raspoložive analize

Odgovarajuća struktura i arhitektura višeslojnih baza podataka kao deo sistema podrške marketing odlučivanja ima veliki značaj u donošenju poslovnih odluka. Marketing istraživanja obezbeđuju podatke koji su "ulaz" za sistem podrške odlučivanju. Tako se "obrađeni" podaci, u razumljivoj i prihvatljivoj formi prosleđuju donosiocima odluka u preduzeću.

Literatura

1. Eric A. S., Active Server Pages 3, Micro knjiga Hungry Minds, Inc., Beograd, 2001. godina.
2. Loney K., Oracle database 10g, "Kompjuter biblioteka", Čačak, 2004. godina
3. Mogin P., Luković I., Govedarica M., Principi projektovanja Baza podataka, Fakultet tehničkih nauka, Novi Sad, 2004. godina.
4. Marčićević Ž., Primena računara, Fakultet za sport i turizam – Tims, Novi Sad, 2007. godina.
5. Radulović B., Informacioni sistemi, Tehnički fakultet "Mihajlo Pupin" Zrenjanin, 2006. godina.
6. Veljović V. A., Menadžment informacioni sistemi, "Kompjuter biblioteka", Čačak, 2002. godina.
7. Milisavljević M., Marketing, Savremena administracija, jedanaesto izdanje, Beograd, 1990. godina, str. 69.

Momčilo Đorđević, Željko Marčićević

THE STRUCTURE OF THE DISTRIBUTED MULTILAYER DATABASES

Abstract: The system of support for marketing decision-making includes a coordinated collection of data, systems, means and technique, together with the supporting hardwares and software target at gathering, analysis, interpretation and presentation of information for those who need to make marketing decisions. That facilitates better understanding for managers, and better solutions for problems at the same time. The paper deals with the concept, structure and the architecture of multilayer databases in order to perform a better analysis and better implementation of the information system model. The new architecture of the multilayer databases imposes itself as a base for realizing the distributed databases on the partitioned Internet area.

Key word: Data Base, Architecture, Internet.