Posrednici umreženih sustava

Prof.dr.sc Siniša Srbljić

Dr.sc. Ivan Benc

Dr.sc. Daniel Skrobo

Fakultet elektrotehnike i računarstva,
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

7. Predavanje

Posrednici za pristup podacima i izvođenje transakcija

Marin Šilić, dipl.ing.

Fakultet elektrotehnike i računarstva
Laboratorij za potrošaču prilagođeno računarstvo

Sadržaj

Uvod

- **Transakcije**
  - Pristup za modeliranje i izgradnju sustava
    - Pouzdani
    - Otporni na pogreške

Uvod

- **Osnovna zamisao transakcije**
  - Preoblikuje sustav iz jednog konzistentnog stanja u drugo konzistentno stanje
  - Niz operacija do konačnog konzistentnog stanja
  - U međukoracima sustav može biti nekim nekonzistentnim međustanjima

Uvod

- **Primjer transakcije**
  - Prijenos sredstava sa jednog bankovnog računa na drugi
  - Dvije logičke operacije
  - Skidanje sredstava s jednog računa
  - Dodavanje sredstava na drugi račun
  - Ne smije doći do nekonzistentnog stanja (npr. jedna od dvije operacije se ne izvrši)

Uvod

- **Transakcije - nije trivijalno, brojne poteškoće**
  - BP – trajni kvarovi medija za pohranu
  - Višeprocesorski sustavi – istovremeno izvođenje transakcija
  - RS – prekid komunikacije, dostupnost udaljenih komponenti
  - Internet – otvoreni i raznorodni sustavi (dodatne poteškoće)

Uvod

- **Izazovi za podršku transakcijama – nadogradnja**
  - Programski jezici
  - APIs
  - Business process modeling languages
  - Komunikacijske posrednike

Uvod

- **Transakcijski posrednici**
  - Transaction Processing Monitors (TP Monitors)
    - Prve arhitekture za nadzor izvođenja transakcija
  - Transaction Processing Middleware (TPM)
    - Razvili se iz TP Monitors

Uvod

- **Uloga transakcijskih posrednika**
  - Olakšati izgradnju i postavljanje pouzdanih, transakcijskih primjenskih sustava koji podržavaju razmjeran rast
  - Skriva se i izbjegava upravljanje transakcijama na niskoj razini
  - Koriste se usluge posrednika tijekom pristupa dijeljenim resursima
  - Programer se može fokusirati na ostvarenje poslovne logike primjenskih sustava


THE UNITED STATES OF AMERICA
DE14835671A

Osnove o provođenju transakcija

- **ACID model transakcija**
  - Atomarnost (Atomicity)
    - Nedjeljiva operacija
  - Konzistentnost (Consistency)
    - Sustav ide iz jednog konzistentnog stanja u drugo konzistentno stanje
  - Izolacija (Isolation)
    - Druge transakcije koje se možda izvode konkurentno nemaju utjecaj na krajnji ishod
  - Trajnost (Durability)
    - Jednom potvrđena transakcija ostaje trajna čak i u slučaju rušenja sustava

Atomarnost

- **Dva ishoda transakcije**
  - Transakcije je uspjela
    - Efekt transakcije se reflektira na stanje sustava
  - Transakcije nije uspjela
    - Operacije transakcije se poništavaju a sustav se vraća u inicijalno konzistentno stanje u kojem je bio prije početka izvođenja transakcije
  - Atomarnost se postiže two-phase commit (2PC) i DO- UNDO-REDO protokolima

Atomarnost

- **2PC protokol**
- **Transakcije u raspodijeljenim sustavima**
- **U sustavu postoje**
  - Resource Manager (RM), upravlja sredstvima
  - Transaction Manager (TM), upravlja transakcijama
- **Prva faza**
  - TM traži od svih RM u sustavu dopuštenje za potvrđivanje transakcije
- **Druga faza**
  - Nakon što svi RM dopuste potvrđivanje u drugoj fazi TM potvrđuje transakciju

Atomarnost

- **DO-REDO-UNDO**
- **Transakcije u centraliziranim sustavima**
- **Sve operacije su ostvarene kao logičke funkcije**
- **Sve funkcije se zapisuju u dnevnike transakcije**
  - DO (Izvođenje transakcije)
  - UNDO (Rollback, poništavanje transakcije)
  - REDO (Recovery, Obnavljanje transakcije)

Konzistentnost

- **Transakcija**
  - Preoblikuje sustav iz jednog u drugo konzistentno stanje
  - Nema utjecaja istodobno izvođenje više transakcija na ishod pojedine transakcije
  - Programer piše transakcije kao individualne logičke cjeline

Izolacija

- **Višeprocesorski sustavi**
- **Izvođenje više transakcija istovremeno**
- **Problemi**
  - DIRTY READS
  - NONREPEATABLE READS
  - PHANTOMS

Izolacija

- **DIRTY READS**

id	name	age
1	Joe	20
2	Alice	25

Transakcija A

```
/* Query 1 */
SELECT * FROM users
WHERE id = 1;
```

```
/* Query 1 */
SELECT * FROM users
WHERE id = 1;
```

Transakcija B

```
/* Query 2 */
UPDATE users SET age = 21
WHERE id = 1;
/* No commit here */
```

```
ROLLBACK; /*
lock-based DIRTY
READ */
```

Izolacija

- **NON-REPEATABLE READS**

id	name	age
1	Joe	20
2	Alice	25

Transakcija A

```
/* Query 1 */
SELECT * FROM users WHERE
id = 1;



Query 1 */
SELECT * FROM users WHERE
id = 1;
COMMIT;
/* lock-based REPEATABLE
READ */
```

Transakcija B

```
/* Query 2 */
UPDATE users SET age = 21 WHERE
id = 1;
COMMIT;
/*lock-based READ COMMITTED */
```

Izolacija

- **PHANTOMS primjer**

id	name	age
1	Joe	20
2	Alice	25

Transakcija A

```
/* Query 1 */
SELECT * FROM users WHERE
age BETWEEN 10 AND 30;
```

```
/* Query 1 */
SELECT * FROM users WHERE
age BETWEEN 10 AND 30;
```

Transakcija B

```
/* Query 2 */
INSERT INTO users VALUES
( 3, 'Bob', 27 );
COMMIT;
```

Izolacija

- **Two-phase-locking protocol (2LP)**
  - SHARED LOCK (S) za čitanje
  - EXCLUSIVE LOCK (X) za pisanje
- **Faza 1**
  - Transakcija mora zatražiti i dobiti sve potrebne ključeve za izvođenje svojih operacija
- **Faza 2**
  - Nakon što se izvrše sve operacije transakcije, transakcija otpušta sve ključeve
- **S ključ je u konfliktu s X ključem**
- **X ključ je u konfliktu s X ključem**

Trajnost

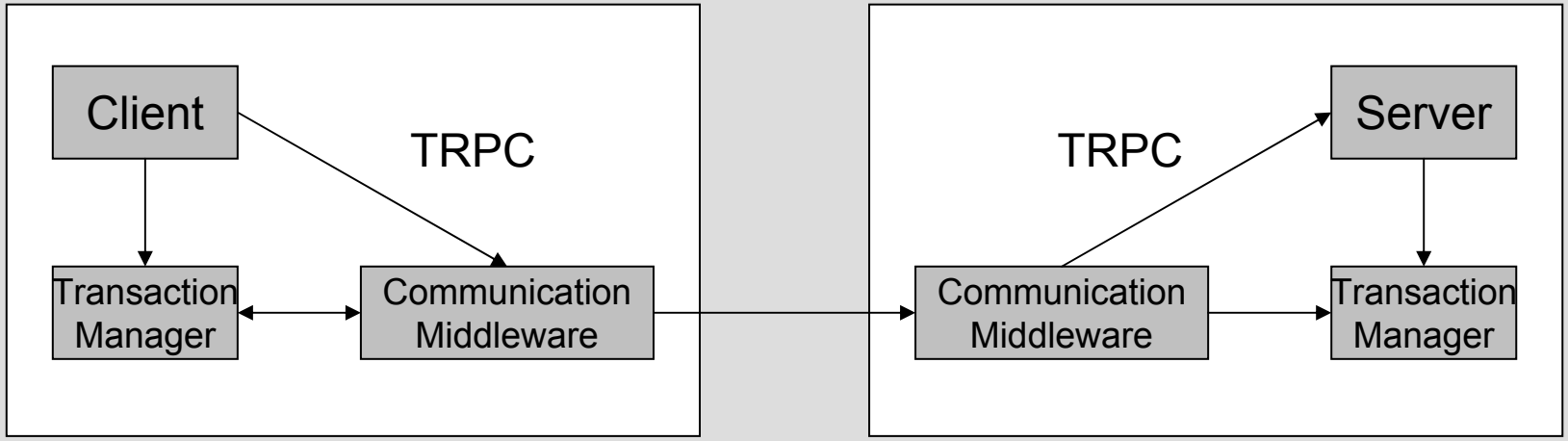
- **Sve promjene/efekti potvrđene transakcije moraju ostati trajne**
- **DO-UNDO-REDO dnevnici transakcija**
- **Write-ahead log rule**
- **U slučaju kvara**
  - REDO svih potvrđenih transakcija
  - UNDO svih otkazanih transakcija
  - DO svih nezavršenih transakcija

Raspodijeljene transakcije

- Teže je ostvariti ACID model
- Više nazavisnih čvorova sudjeluje u transakcijama
- Resursi su raspoređeni na različite čvorove
- Problemi u komunikaciji
- Kvar nezavisnog čvora

Raspodijeljene transakcije

- **Transactional RPC**
  - Nadogradnja na RPC
  - Omogućuju klijentu izvršavanje operacija na udaljenim resursima



Raspodijeljene transakcije

- **Raspodijeljeni Two-Phase Commit**
- **Koordinator mora čekati na odgovor svih sudionika**
- **Što ako se sruši pojedini lokalni RM, ili što ako se sruši TM?**
  - Donose se heurističke, ad-hoc odluke
  - Može doći do narušavanja atomarnosti i konzistencije, ponekad je potrebna intervencija čovjeka da se sustav dovede u ispravno stanje

ACID poteškoće

- **ACID ponekad nepraktičan za neke aplikacije**
  - Pristup s ključem pogodan za kratke transakcije
  - Nepraktičan za duge transakcije
  - UNIT of WORK i UNIT of failure
    - Svi efekti neuspješne transakcije se poništavaju
  - Nema modularnosti kao drugim programskim jezicima
    - Ne dopuštaju se ugnježdene transakcije
  - Uvode se brojne varijacije klasičnog ACID modela

ACID varijacije

- **Razine Izolacije**
  - Neke aplikacije (npr. koje samo čitaju) dopuštaju međudjelovanje prilikom istovremenog izvođenja
  - Serializable
    - Fantomi nisu dopušteni
  - Repeatable Read
    - Fantomi dopušteni
  - Read Commited
    - Fantomi i nonrepeatable reads dopušteni
  - Read Uncommited
    - Fantomi, nonrepeatable i dirty reads dopušteni

ACID varijacije

- **Optimističan nadzor istovremenog izvođenja transakcija**
  - Svaka transakcija dobije svoje kopije podataka
  - Tijekom potvrđivanja se razriješavaju konflikti ukoliko do njih dođe
- **Ugnježđene transakcije**
  - Podtransakcije
    - Dobiva ključ ako je slobodan ili ga ima roditelj
    - Kad se potvrdi podtransakcija, roditelj dobiva ključeve koje je ona držala
    - Kad se otkaže podtransakcija, ključevi se oslobađaju

ACID varijacije

- **Compensation-based recovery**
  - Duge transakcije se razbijaju na niz kraćih ACID transakcija
  - Sve skupa čine logički ACID transakciju, ali rezultati svake kraće ACID transakcije su vidljivi drugim logičkim ACID transakcijama
  - U slučaju rollback, onda se izvršava niz kompenzacijskih ACID transakcija


THE UNITED STATES OF AMERICA
DE14835671A

Objektno-orijentirani transakcijski posrednici

- **Distributed object transaction**
  - Razvio se iz RPC
  - Uvodi se objektno-orijentirani pristup za izgradnju raspodijeljenih sustava

Objektno-orijentirani transakcijski posrednici - klijent

- **Object Transaction Service (OTS) standard**
  - Eksplicitan API za transakcije
  - Grupiranje logike transakcije u programske blokove
  - Korištenje ključnih riječi za početak i zavretak transakcije
    - Begin
    - Commit
    - Rollback

Objektno-orijentirani transakcijski posrednici - klijent

- **J2EE implementacija OTS sučelja**
  - *javax.transaction.UserTransacation* interface

```
UserTransaction ut = new UserTransaction();
try
{
    ut.begin();
    // Logika transakcije
    ut.commit();
}
catch(Exception e)
{
    ut.rollback();
}
```

Objektno-orijentirani transakcijski posrednici - poslužitelj

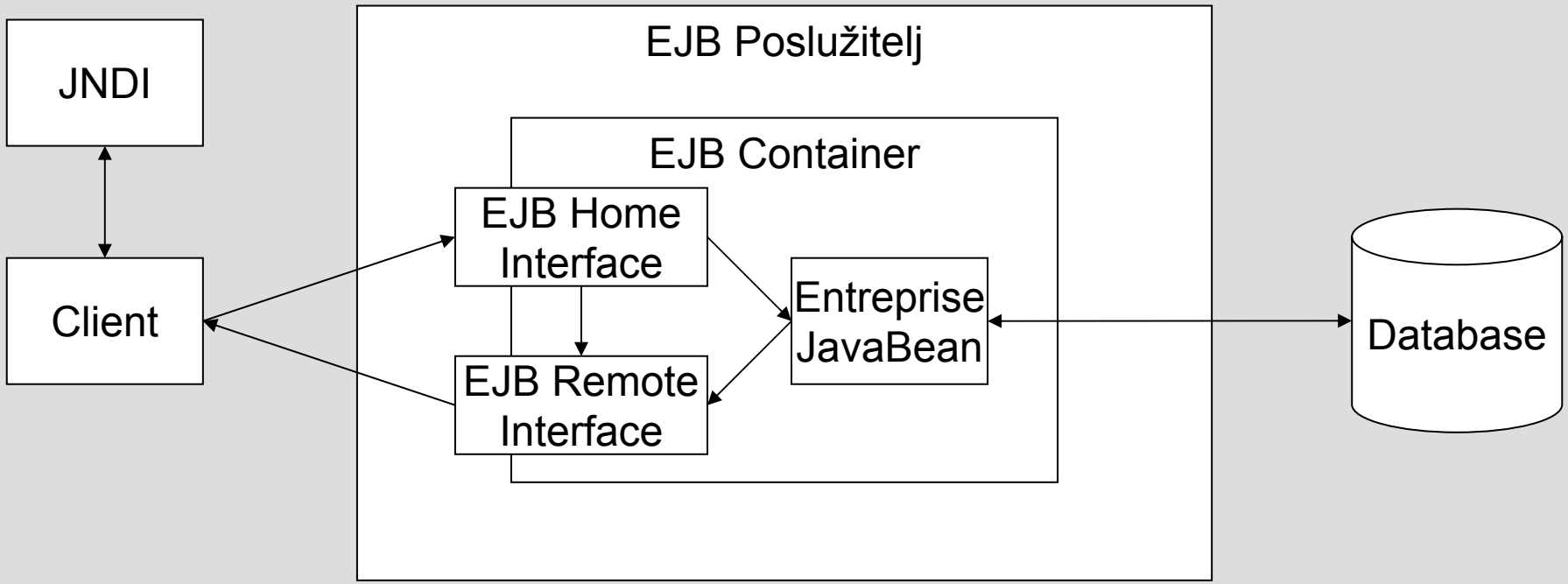
- **Kako ostvariti transakcijske poslužitelje?**
- **J2EE podržava više modela za ostvarenje i postavljanje transakcijskih poslužitelja**
  - Enterprise JavaBeans (EJB)

Entreprise JavaBeans

- **Standard za izgradnju poslovnih aplikacija**
- **Poslužiteljski kod – isti izazovi i zahtjevi**
  - Postojanost podataka
  - Integritet transakcija
  - Istodobno izvođenje
  - Sigurnost
- **Ideja je razviti okolinu gdje se programeri mogu fokusirati na logiku samog primjenskog sustava**
- **EJB se brine o navedenim zahtjevima**

Entreprise JavaBeans

- **Arhitektura sustava EJB – container managed transactions**



Entreprise JavaBeans

- **Postavljanje**
  - Opisnik za postavljanje (Deployment descriptor) – XML dokument koji sadrži sljedeće informacije za svaki EJB koji se želi postaviti
    - Ime Home sučelja
    - Java razred za Bean (objekt koji sadrži logiku)
    - Java sučelje za Home sučelje
    - Java sučelje za Bean
    - Sigurnosne postavke i prava pristupa
    - Transakcijske postavke (Razina izolacije, kontekst transakcije)


THE UNITED STATES OF AMERICA
DE14835671A

Transakcije u okolinama zasnovanim na porukama

- **RPC model nedostatci**
  - Klijent i poslužitelj dostupni istodobno
  - Klijent blokiran dok ne dobije odgovor od poslužitelja
  - Čvrsto povezan sustav, nije uvijek ostvariv
  - Klijent možda želi obraditi zahtjev i ako poslužitelj ne odgovara
  - Klijent možda želi poslati zahtjev grupi poslužitelja a ne samo jednom

Transakcije u okolinama zasnovanim na porukama

- **Message-oriented middleware MOM (Posrednici zasnovani na porukama)**
  - MOM je posrednik između aplikacija
  - Aplikacije komuniciraju međusobno preko MOM
  - MOM pohranjuje poruke i odgovore
  - MOM jamči isporuku zahtjeva ili odgovora
  - Asinkroni komunikacijski model
    - *Fire and forget a message*
  - Aplikacije ne moraju biti dostupne u istom trenu
  - Aplikacija nije blokirana i ne čeka ni u jednom trenu

Transakcije u okolinama zasnovanim na porukama

- **Modeli komunikacije zasnovane na porukama**
  - Point-to-point messaging (komunikacija preko Reda poruka)
  - Publish/subscribe messaging (objava/pretplata model komunikacije)

Point-to-point messaging

- Komunikacija preko *Redova poruka*
- Posrednik između aplikacija koje komuniciraju
- Red poruka je postojan spremnik za poruke
- Red poruka ima svoj logički identifikator i njime upravlja Upravitelj Reda poruka (Queue Manager)
- Redovi poruka se mogu koristiti kao transakcijski resursi, tada svaka operacija pisanja u ili čitanja iz Reda poruka ovisi o sudbini transakcije unutar koje se izvodi

Publish/subscribe messaging

- **Objava/pretplata model komunikacije**
  - Proizvođači poruke objavljuju na određene teme
  - Potrošači se pretplate na određene teme i dobijaju sve poruke koje se objave na te teme
  - Na ovaj način poruku može dobiti više različitih potrošača za razliku komunikacije preko Reda poruka

Programski modeli

- **J2EE Java Message Service**
  - Standardni API
  - Definira sučelja za
    - Point-to-point komunikaciju
    - Publish/subscribe komunikaciju
  - Uvodi se *transcated session*
    - Sve poruke unutar transakcijske sjednice postaju dio transakcije
    - Poruke se šalju i primaju jedino ako se transakcija potvrdi

Programski modeli

- **J2EE Java Message Service – transacted session**

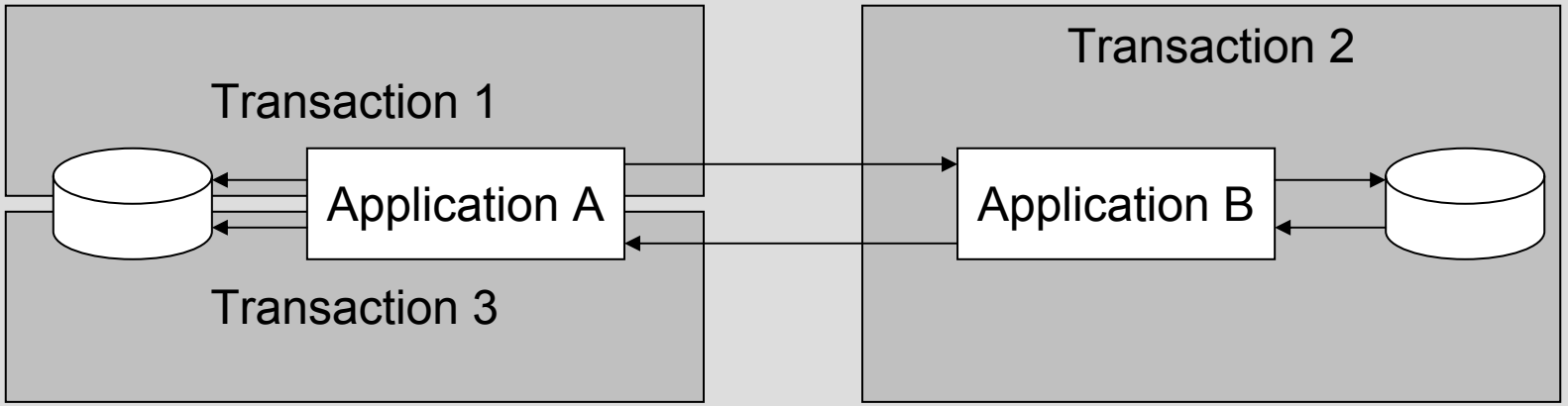
```
QueueSession qs = connection.createQueueSession(true,
    Session.AUTO_ACKNOWLEDGE);

QueueReceiver receiver = qs.createReceiver(queue);
TextMessage t1 = (TextMessage) receiver.receive();

QueueSender sender = qs.createSender(queue);
TextMessage t2 = qs.createTextMessage();
t2.setText(text);
sender.send(t2);

session.commit();
```

Programski modeli

- **Message-driven Bean (MDB)**
  - Integracija JMS i EJB
  - MDB je Bean u Containeru i pridružen je nekom Redu poruka ili nekoj Temi
  - U MDB opisniku je definirana metoda koju je potrebno pozvati i predati joj pristiglu poruku
  - EJB presreće JMS poruke i poziva odgovarajuće metode

Queued Transaction Processing

- **Predložak koji se koristi u brojnim primjenskim sustavima zasnovanim na porukama**
- **Klijent i poslužitelj komuniciraju koristeći MOM i slijed izravnih raspodijeljenih transakcija**




THE UNITED STATES OF AMERICA
DE14835671A

Transkacije na Web-u

- **Web Services tehnologija**
  - Raspodijeljeni programski model
  - Arhitektura zasnovana na uslugama (SOA)
  - Standardi WSDL, UDDI i SOAP
- **Potreba za pouzdanim transakcijama i komunikacijom**
  - WS-Coordination
  - WS-Transaction
  - WS-ReliableMessaging
- **Service oriented middleware (SOM)**
  - Upravlja raspodijeljenim, decentraliziranim resursima na Web-u

Transkacije na Web-u

- **WS-Coordination**
  - Opisuje radni okvir za koordinaciju Web transakcija
  - Stvaranje konteksta transakcije
  - Razmjenu konteksta među uslugama sudionicima
- **WS-Transaction**
  - Atomic Transaction (AT)
    - Kraće transakcije
    - Unutar povjerljive domene
  - Bussines Activity (BA)
    - Duge transakcije
    - Unutar različitih domena

Programski model

- **Bussines Process Execution Language for Web Services (BPEL)**
  - Jezik za definiciju, koordinaciju i izvođenje poslovnih procesa izgrađenih od Web usluga
  - Tok podataka (Data Flow)
  - Tok upravljanja (Control Flow)
  - Definicija i izvođenje poslovnih transakcija
  - Rukovanje iznimkama
- **Standardiziran od OASIS**


THE UNITED STATES OF AMERICA
DE14835671A

Napredne transakcije

- **Long Running Unit of Work (LRUOW)**
  - Model omogućuje istodobno izvođenje dužih transakcija bez zaključavanja pojedinih sredstava
    - Packaging Control – Poslovne aktivnosti grupira u Jedinice posla (Unit of Work, UoW)
    - Visibility Control – Pojedini objekti koji su stvoreni ili mijenjani su vidljivi samo unutar određenog konteksta
    - Concurrency Control – omogućuje se procesima da pristupaju istim podacima

Napredne transakcije

- **Long Running Unit of Work (LRUOW)**
  - LRBP se modelira kao aciklički usmjereni graf čiji su čvorovi Jedinice posla (UoW)
  - Svaki čvor ima jednog roditelja i može imati više djece
  - Svaki podzadatak u poslovnom procesu se vezuje na jedan čvor u grafu i izvodi se u tom kontekstu
  - Moguće je istodobno izvođenje nad istim podacima, pritom svaki proces dobija svoju kopiju podataka i objekata, te se sve bilježi u čvoru roditelju
  - Jedinica posla se potvrđuje ili opoziva izvođenjem odgovarajuće metode nad čvorom
  - Svaki pojedini UoW se izvodi u dvije faze
    - Long-running phase (izvođenje samog posla)
    - Short-running phase (razrješuju se posljedice istodobnog izvođenja)

Napredne transakcije

- **Conditional Messaging (Komunikacija porukama uz uvjet)**
  - Nadogradnja na standardni MOM
  - Omogućuje definiciju različitih uvjeta kao nezavisnih objekata o kojima ovisi isporuka i obrada poruka
  - Omogćuje nadzor isporuke ili obrade poruke primatelju slanjem poruke o potvrdi primitka ili završetka obrade
  - Omogućuje izvrijednjavanje uvjeta zbog utvrđivanja uspješnog ili neuspješnog slanja/obrade poruke
  - Omogućuje provođenje određenih akcija u ovisnosti o uspješnoj ili neuspješnoj obradi poruke, npr. slanje potvrdne obavijesti u slučaju uspjeha ili slanjem kompenzacijskih poruka u slučaju neuspjeha
  - Komunikacija porukama uz uvjet pomiče odgovornost o uvjetima isporuke i obrade poruka iz primjenskog sustava u posrednički sustav

Zaključak

- Obrada transakcija je ključna za razvoj bilo kojeg poslovnog primjenskog sustava
- Komunikacijski posrednici bilo OOM, MOM, SOM svi podržavaju obradu transakcija
- Prikazana su osnovna svojstva i zahtjevi transakcija
- Napredni transakcijski posrednici (LRUOW, Conditional Messaging)