

Uvod u integraciju poslovnih aplikacija (Enterprise Application Integration - EAI)

Kada različiti sistemi ne mogu da efikasno dele i razmenjuju podatke, u informacionom sistemu se stvaraju uska grla kroz koje se protok ne može obezbediti bez ljudske intervencije u obliku donošenja odluka ili unosa podataka. EAI predstavlja disciplinu koja je nastala zahvaljujući godinama razvoja aplikacionih sistema za podršku poslovanju, tokom kojih se nije vodilo računa o sposobnosti tih sistema da međusobno komuniciraju.

Integrisanje poslovnih aplikacija (EAI – Enterprise Application Integration) predstavlja disciplinu koja obuhvata skup procesa, softverskih i hardverskih alata, metodologija i tehnologija, čijom se integrisanom implementacijom postiže konsolidacija, povezivanje i organizovanje svih poslovnih računarskih aplikacija, podataka i procesa u integrisano okruženje koje obezbeđuje razmenu, upravljanje i reformulaciju informacija i znanja iz domena poslovanja preduzeća, u realnom vremenu.

Integrisano rešenje se razvija kreiranjem tzv. EAI zona – grupa aplikacija slične funkcije, odnosno grupa aplikacija čijim zajedničkim funkcionisanjem se obezbeđuje ostvaranje jednog poslovnog cilja. Svaka od pojedinačnih aplikacija sa ostalima u okviru svoje EAI zone treba da komunicira putem jedinstvenog protokola, pri čemu je poželjno da se taj protokol koristi i za komunikaciju između različitih EAI zona. Svaku od aplikacija unutar EAI rešenja treba da karakteriše nezavisnost – interfejs integrisanja predstavlja samo jedan programski sloj nad tom aplikacijom, a eventualne izmene aplikacije, odnosno njena kompletna zamena drugom ne sme da utiče na funkcionalnost ukupnog integrisanog okruženja.

Osnovne karakteristike

Osnovne karakteristike aplikacionog okruženja integrisanog primenom EAI principa su:

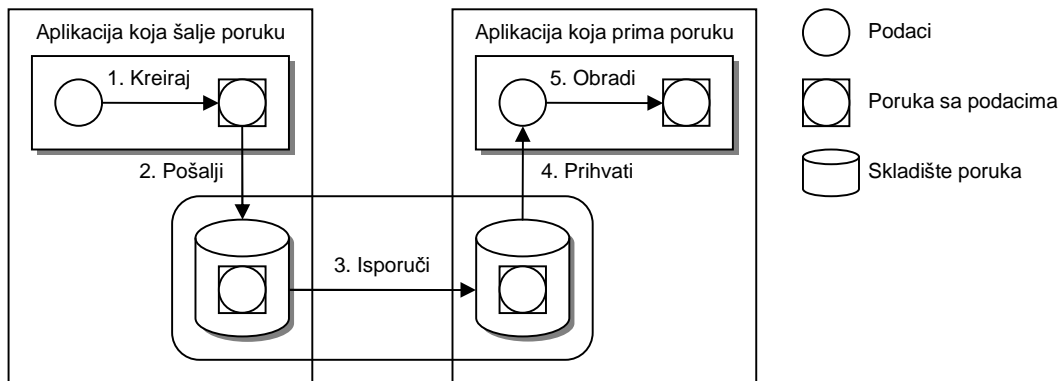
- Interoperabilnost
- Sinhrona i asinhrona interakcije
- Preslikavanje i transformacija podataka
- Procesna logika (deterministička i nedeterministička)
- Kompenzacija transakcija

Interoperabilnost aplikacija

Interoperabilnost jedne aplikacije u odnosu na njeno okruženje karakteriše sposobnost slanja odgovarajuće poruke odgovarajućoj aplikaciji, bez obzira na to gde se nalazi ta aplikacija, da li je aktivna ili ne i, bez obzira na to u kom okruženju (operativni sistemi, hardver) funkcioniše.

Interoperabilnost aplikacija može da se sagleda sa dva aspekta, od kojih svaki karakterišu različiti principi. Prvi se odnosi na internu komunikaciju dve aplikacije, unutar granica poslovnog sistema. Drugi je sposobnost integrisanog informacionog sistema, odnosno odgovornih aplikacija, da komunicira sa spoljnim svetom – aplikacijama koje se nalaze izvan granica poslovnog sistema.

Osnovni nosioci komunikacije između različitih aplikacija u EAI okruženju se nazivaju porukama (message), pri čemu se interakcija dve aplikacije ostvaruje njihovom razmenom (messaging). Razmena poruka se ostvaruje putem odgovarajućih kanala (channels), duž kojih se kreću poruke – paketi podataka jedinstvene, definisane strukture. Poruka se sastoji iz zaglavlja (header), u kojem se nalaze identifikacije pošiljaoca i primaoca poruke i drugi meta podaci, i tela (body), koje čini sam sadržaj poruke.



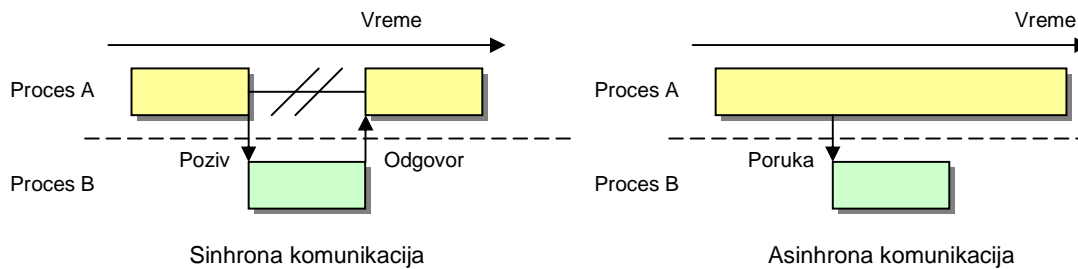
Na slici je prikazana struktura interakcije dve aplikacije u EAI okruženju, putem razmene poruka. Ona se obavlja u 5 koraka:

1. Kreiranje poruke. Aplikacija – pošiljalac kreira poruku i popunjava podatke koji čine njen sadržaj;
2. Slanje poruke. Aplikacija – pošiljalac šalje poruku u kanal komunikacije. Nakon slanja poruke, aplikacija – pošiljalac nastavlja sa radom, pri čemu nema više nikakvih odgovornosti u procesu razmene poruka;
3. Isporučka poruke. EAI sloj za interakciju koji implementira kanal komunikacije se stara o tome da se poruka prenese duž mrežne infrastrukture, sa računara na kome se izvršava aplikacija – pošiljaoca, na računar na kojem se izvršava aplikacija – primaoca poruke.
4. Prijem poruke. EAI sloj za interakciju se stara o tome da je poruka primljena, skladištena na računar na kome se izvršava aplikacija – primaoca, i dostupna njoj u svakom ili odgovarajućem trenutku;
5. Obrada poruke. Aplikacija – primalac vrši ekstrakciju podataka iz poruke i koristi ih za obradu.

Sinhrona i asinhrona interakcije

Sinhrona interakcija predstavlja razmenu poruka koja se obavlja u realnom vremenu, pri čemu aplikacija koja šalje poruku drugoj aplikaciji, očekuje od nje odgovor u realnom vremenu. Komunikaciju karakteriše dvosmerni tok informacija u jednom komunikacionom kanalu, duž kojeg se kreću zahtev (request) i njegov odgovor (response).

Ipak, veliki broj poslovnih okolnosti karakterišu problemi zbog kojih se sinhronom komunikacijom ne može zaokružiti jedan poslovni proces na pouzdan i efikasan način. Naime, u velikom broju slučajeva, aplikacija koja upućuje zahtev mora da čeka na njegov odgovor, zbog vremena potrebnog za procesiranje zahteva, proisteklog iz potrebe za reakcijom operatera ili zbog opterećenja računarskih resursa. Ovo stanje čekanja blokira rad aplikacije koja je uputila poziv i zauzima mrežne resurse. Sa druge strane, nepouzdanost i sporost komunikacione infrastrukture može da uvede veliki rizik narušavanja integriteta poslovnog sistema usled prekinute sinhrona interakcije aplikacija.



Asinhrona komunikacija se ostvaruje u dva komunikaciona kanala, duž kojih se kreću zahtev i odgovor u različitim smerovima. Ona se ne vrši u realnom vremenu, već u konačno dugom vremenskom periodu, pri čemu se dobijanje odgovora obezbeđuje novim zahtevom ili otvaranjem kanala za prijem odgovora, koji funkcioniše uporedno u odnosu na ostale funkcije aplikacije i ne angažuje značajne aplikacione resurse.

Preslikavanje (mapping) i transformacija podataka

Aplikacije u jednom heterogenom okruženju nisu projektovane, razvijene i instalirane sa ciljem da funkcionišu integrisano. One uobičajeno koriste različite formate podataka i protokole interfejsa za eksternu komunikaciju. Zbog toga, neophodno je da EAI sistem karakteriše sposobnost preslikavanja i transformacije podataka u realnom vremenu, prilikom njihovog transfera i njihovog preslikavanja iz izvornog u ciljani format.

Preslikavanje podataka (data mapping) podrazumeva uspostavljanje referenci između formata podataka koji se koristi za njihovo skladištenje (najčešće, relacioni model) i formata podataka koji se koristi za obradu unutar jedne aplikacije.

Transformacija podataka predstavlja aktivnost konverzije podataka u prihvatljive formate sa stanovišta njihovog korisnika. Osnovna karakteristika uobičajene strategije integriranja informacionog sistema preduzeća je korišćenje jedinstvenog – kanoničnog formata podataka u njihovom transportu od jedne do druge aplikacije. Kanoničan model podataka je model koji je nezavisan od bilo koje aplikacije u okruženju. Danas, najkorišćeniji kanoničan model podataka je XML (eXtensible Markup Language).

Procesna logika (deterministička i nedeterministička)

Poslovne informacione sisteme karakteriše složena poslovna logika. EAI sistem treba da karakteriše sposobnost izražavanja i automatizacije ove logike i to primenom paradigme poslovnih procesa, odnosno radnih tokova (workflow).

Kompenzacija transakcija

S obzirom na zahtev za asinhronom komunikacijom unutar jednog integrisanog poslovnog sistema, od izuzetne je važnosti da se očuva integritet podataka koji se razmenjuju u toku jedne interakcije dve aplikacije, odnosno integritet same interakcije. Osnovno sredstvo za očuvanje integriteta jedne interakcije je njena transakciona priroda.

Transakciona interakcija dve aplikacija u okviru EAI okruženja je izvršena samo i jedino nakon eksplicitne potvrde (commit) njenog izvršenja. Pre potvrde, interakcija se može otkazati, bez obzira na fazu u kojoj se nalazi njeno izvršenje. Otkazivanje transakcione interakcije nakon potvrde nije moguće ni u kom slučaju. U slučaju da je izvršenje jedne interakcije dovelo aplikacioni sistem u nestabilno ili generalno, neželjeno stanje, on se može povratiti u stabilno stanje primenom kompenzacije transakcije.

Kompenzacija transakcione interakcije obuhvata njeno zaustavljanje i reviziju definisanog, već izvršenog skupa operacija transfera i obrada. Iako se često pretpostavlja da je uloga kompenzacije povraćaj aplikacionog sistema u prethodno stanje, to nije uvek tačno –

kompensacija može da podrazumeva i dovođenje sistema u balansirano i konzistentno stanje, umanjnjem ili poništavanjem dejstva interakcije ili interakcija koje su ga dovele u neželjeno stanje.

Osnovni principi implementacije

EAI predviđa različite načine implementacije, od kojih su najčešće korišćeni:

1. **Integrisanje informacija.** Obezbeđivanje konzistentnosti informacije koja se koristi u različitim sistemima.
2. **Procesna integracija.** Projektovanje procesa koji prožimaju rad različitih aplikacionih sistema
3. **Vendor independence.** Ekstrahovanje poslovne logike iz aplikacionih sistema i njihovo implementiranje u EAI sistem, tako da i u situacijama kada se aplikacioni sistem zameni drugim sistemom različitog vendara, poslovna logika ne mora da se ponovo implementira.
4. **Fasadni pristup.** EAI sistem igra ulogu front-enda - fasade za grupu aplikacija, obezbeđujući jedinstven i konzistentan interfejs za sve aplikacione sisteme.

Osnovni EAI Paterni

Osnovni paterni za integrisanje poslovnih aplikacija su medijacija i federacija. Iako ih karakterišu potpuno različiti koncepti, u praksi se veoma često primenjuju uporedno.

Patern medijacije podrazumeva da se EAI sistem se koristi kao broker komunikacije različitih aplikacija. Kada god dođe do promene u jednom sistemu (unos nove informacije, završetak transakcije i sl.), integracioni modul EAI sistema se o tome obaveštava. Njegova uloga je da propagira izmene o kojima je obaveštenem drugim, relevantnim aplikacijama.

Patern federacije karakteriše uloga fasade, koju EAI sistem ima, u odnosu na aplikacije čiji rad integriše. U ovom slučaju, svi pristupi poslovnim aplikacijama, koji se nalaze u zoni integracije se vrše preko front-enda koji obezbeđuje EAI sistem. On je, u ovom slučaju konfigurisan tako da korisniku prezentuje i obezbedi pristup svim relevantnim informacijama i funkcijama aplikacija i, na osnovu korisnikovih aktivnosti ostvari odgovarajuću interakciju sa poslovnim aplikacijama.

Middleware tehnologija za realizaciju EAI sistema

Middleware predstavlja osnovnu tehnologiju realizacije jednog EAI sistema, pri čemu se veoma često koristi i za podršku funkcionisanju kompleksnih distribuiranih aplikacija.

Middleware predstavlja bilo koji softver ili deo softvera koji povezuje dve ili više aplikacija na način koji obezbeđuje razmenu podataka. Iako se termin koristi još od 1968 (NATO Software Engineering Conference), ova tehnologija je stekla popularnost osamdesetih godina, kao rešenje za povezivanje savremenih aplikacionih sistema sa legacy softverom. U poslednje vreme, zahvaljujući razvoju distribuiranih softverskih sistema, middleware tehnologije dobijaju značajnu pažnju. U tom smislu, jedna od najznačajnijih primena middleware koncepta, danas su aplikacioni serveri.

Neke od značajnih primena middleware koncepta su sledeće tehnologije:

- Poziv udaljene procedure (Remote Procedure Call - RPC). Tehnologija omogućava da klijentski softver vrši pozive procedura koje se izvršavaju na udaljenom računaru – serveru.
- Message Oriented Middleware (MOM). Softver koji omogućava asinhonu komunikaciju između različitih aplikacija korišćenjem poruka kao formata za slanje i prijem podataka koje aplikacije razmenjuju.
- Object Request Broker (ORB). Tehnologija koja omogućava razmenu podataka između dva objektno-orijentisana sistema, i to u formi objekata.
- SQL-orijentisan pristup podacima. Deo softvera koji se koristi za razmenu podataka između aplikacije i sistema za upravljanje bazama podataka.
- Aplikacioni serveri. Softver koji se koristi za izvršavanje drugih aplikacija, najčešće distribuiranih.

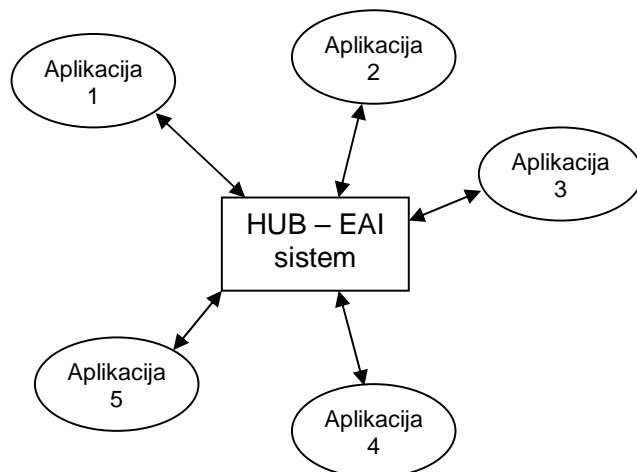
Jedna od najznačajnijih primena middleware koncepta u ovom kontekstu su enterprise service bus-ovi. Njihova definicija, funkcionisanje i primena će biti predstavljeni u okviru narednog poglavlja.

Arhitektura EAI sistema

Postoje dva osnovna pristupa projektovanju EAI sistema: *hub-and-spoke* i *enterprise service bus (ESB)*. Prvi pristup karakteriše centralna pozicija EAI sistema, koji igra ulogu hub-a, posrednika sa aplikacija koje se nalaze u zoni integracije. U drugom pristupu, EAI sistem ima ulogu magistrale, ili je implementiran kao rezidentan modul postojeće magistrale ili middleware-a.

Hub-and-spoke arhitektura EAI sistema

Hub-and-spoke arhitektura predstavlja realizaciju topologije zvezde u oblasti integrisanja poslovnih aplikacija, pri čemu EAI sistem ima centralnu ulogu posrednika u interakciji između bilo koje dve aplikacije iz zone integracije.



Osnovna prednost ovakve topologije je veoma jednostavno dodavanje novih aplikacija u zonu integracije, implementacijom jedinstvenog interfejsa za komunikaciju sa EAI sistemom.

Ipak, hub-and-spoke arhitektura se relativno retko koristi za integrisanje velikog broja aplikacija. Razlozi za to su relativno mala fleksibilnost i uobičajeno visoko opterećenje hub-a:

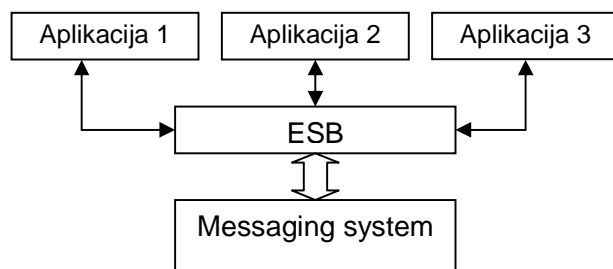
Centralizovani model karakteriše relativno mala fleksibilnost. Izmene u funkcionisanju hub-a mogu dovesti do neočekivanih posledica u komunikaciji sa aplikacijama iz zone integracije. Sa druge strane, periodično veliki saobraćaj na pojedinačnim krakovima, nasuprot malom ili nikakvom opterećenju na drugim krakovima, može značajno usporiti performanse EAI sistema, odnosno, dovesti u pitanje njegovo funkcionisanje.

Hub predstavlja usko grlo ukupne zone integracije. Naime, on je posrednik za sav saobraćaj koji je namenjen integrisanju bilo koje dve aplikacije u zoni integracije. U tom smislu, ukupan kapacitet komunikacije svih aplikacija u zoni integracije je ograničen kapacitetom hub-a. Očigledno je da bilo kakav zastoj u funkcionisanju hub-a, usled njegove preopterećenosti, utiče na ukupne performanse sistema, naročito ukoliko je u projektovanju EAI sistema korišćen patern federacije.

ESB pristup u projektovanju EAI sistema

Enterprise Service Bus predstavlja oblik arhitekture softvera, zasnovan na standardima složenijih event-driven arhitektura i messaging sistema, koji se implementira primenom middleware tehnologija.

Generalno, ESB predstavlja apstraktni sloj *messaging* sistema informacionog sistema preduzeća, postavljen tako da može da koristi njegove funkcije, bez dodatnog razvoja.



ESB obuhvata skup standardnih karakteristika – sposobnosti integrisanog sistema, koje se mogu implementirati na proizvoljan način. Opšte prihvaćene funkcije enterprise service bus-a su:

1. Podrška sinhronim i asinhronim transportnim protokolima za poziv funkcija poslovnih aplikacija.
2. Rutiranje poruka kroz zonu integracije, na osnovu definisanog adresnog prostora, odnosno dodeljenih adresa se vrši na osnovu predviđenih putanja poruka ili u realnom vremenu – na osnovu sadržaja poruka.
3. Medijacija aplikacija u zoni integracije se vrši uz pomoć adaptera, pri čemu su obezbeđene funkcije translacije protokola, odnosno transformacije i prevođenja podataka.
4. Na osnovu predviđene definicije poslovnih procesa, može se vršiti orkestracija aplikacija u zoni integracije. Oko ove funkcije ESB, ne postoji industrijski konsenzus. Dok svi ostali vendori smatraju da je ona važna karakteristika enterprise service bus-a, IBM ovu funkciju realizuje posebnim sistemom – WebSphere Process Server, u odnosu na svoj - WebSphere ESB.
5. Obrada događaja u zoni integracije se vrši primenom mehanizama interpretacije događaja, korelacije podataka, upoređivanje paterna, itd.
6. Sigurnost (enkripcija i digitalno potpisivanje), pouzdanost isporuke poruka i implementirani mehanizmi za transakcije
7. Upravljanje ukupnim sistemom se vrši primenom funkcija monitoringa, auditinga, logovanja, itd.

Pored ovih, osnovnih funkcija ESB, primenu ovakvog oblika arhitektura karakterišu i tehnološki zahtevi, kao što su:

- Korišćenje XML-a, kao standardnog jezika za komunikaciju između aplikacija,
- Primenu transformacionih servisa (XSLT, XQuery) za prilagođavanje formata podataka i vrednosti u različitim kontekstima njihovog korišćenja,
- Validacija poruka, naspram predviđene šeme poruka prilikom njihovog slanja i prijema,
- Uobičajeno se implementira nezavisno od korišćenih operativnih sistema i programskih jezika. Naime, ESB treba da obezbedi ravnopravnu interakciju svih poslovnih aplikacija, bez obzira na programski jezik koji je korišćen za njihov razvoj i platformu na kojoj se one izvršavaju.
- Podrška zadržavanju poruka, ukoliko su poslovne aplikacije kojima su one namenjene trenutno nedostupne, i drugi.

Osnovne prednosti primene enterprise service bus-a u integrisanju poslovnih aplikacija su:

- velika fleksibilnost u implementaciji novih zahteva ukupnog sistema;
- zasnovanost na standardima;
- mogućnost primene u različitim opsezima integracije – veličinama zona integracije;
- značajno umanjena potreba za razvojem prilikom implementacije – ona se primarno zasniva na konfigurisanju ESB-a, i;
- implementacija ESB-a, odnosno njegovih pojedinačnih funkcija i veza se može vršiti bez dodatnog opterećenja poslovnih aplikacija i uz održavanje potpunog kontinuiteta njihovog funkcionisanja.

Ključne mane ovog pristupa su:

- ESB se može primeniti samo u poslovnom informacionom sistemu u kojem postoji messaging arhitektura;
- Iako je osnova za fleksibilnost ovog pristupa tzv. slaba integracija, bez značajnog i pažljivog planiranja implementacije ESB, postoji rizik za uspostavljanje upravo čvrste integracije;
- Za implementaciju, konfigurisanje i administraciju ESB, potrebno je angažovati dodatne ljudske resurse;
- Za efikasnu primenu, neophodno je da u preduzeću postoji uređen sistem procesa i detaljno definisana strategija poslovanja.