

Uvod

U ranim sedamdesetim godinama se počelo raditi na povezivanju informatičkih sistema - nastali su standardni protokoli poput X.25, OSI protokola i TCP/IP-a. Programeri su morali sami programirati aplikacije koje su bile u skladu s odabranim protokolom.

Sredinom sedamdesetih su se počele javljati standardne mrežne aplikacije za slanje i prijem podataka i poruka – npr. X.400, X.500, Netware. Za uredan prijenos su se brinuli gotovi software paketi, no podaci su na prijemnoj strani završili u nekoj datoteci, a za daljnju obradu je opet trebalo pisati posebne programe.

Sredina osamdesetih je početak razvoja klijent-server arhitekture. Došlo je do standardizacije programskih interface-a (API) koji su omogućili da programi na raznim računarima međusobno razgovaraju i razmjenjuju podatke. Programeri na obje strane su se morali dogovoriti o izgledu i značenju poruka.

Jednom dogovoreno se teško mijenjalo - jednom odabrani protokol je bilo gotovo nemoguće mijenjati.

Početak devedesetih je učinjen četvrti korak - objektna tehnologija. Aplikacije sarađuju jedna s drugom, ali svaka govori svojim jezikom. Objektno-orijentisane aplikacije se grade od komponenti (objekata). Sve što jedna komponenta (objekt) treba znati o drugima je njeno sučelje (interface).

Komponenta opisuje samo šta radi i kako je treba koristiti. Kako to radi nije važno. To omogućava da se lako zamijeni cijeli objekt nekim drugim, boljim, a da ostatak sistema to ni ne primjećuje.

Pokušajmo to pojasniti analogijom - svrha telefona je da poveže dvije stranke i omogući im glasovnu komunikaciju. Da bi mogli koristiti telefon, trebate znati telefonski broj osobe koju trebate i kako pritiskati tipke na aparatu. Za korištenje telefona nije nam potrebno znati da li je centrala crossbar ili digitalna, da li je biranje pulsno ili tonsko. Zbog takve organizacije, telekom operator mijenja centrale, tehnologiju, a jedino što korisnici vide je promjena tarife. Korisnici vide promjene tek kad se mijenja nešto u samom načinu upotrebe - kad npr. pozivni broj za Sarajevo postane 033, umjesto 071.

Zbog ovakvog pristupa, objektno-orijentisani sistemi se lakše projektuju i lakše održavaju. Kad se jednom definišu sučelja objekata, razvoj pojedinih objekata mogu raditi različite ekipe, paralelno. Gotove komponente se povezuju tek u trenutku same obrade.

Objektna tehnologija ne zahtijeva razvoj od samog početka – postoje razne tehnike koje omogućavaju da se kompletne postojeće aplikacije pretvore u komponente.

Arhitekturu CORBA (Common Object Request Broker Architecture) je razvila OMG (Object Management Group), sa osnovnim zadatkom da projektuje standardnu i portabilnu arhitekturu namijenjenu za razvoj objektno-orijentisanih aplikacija, implementiranih na različitim hardverskim i softverskim platformama.

CORBA je objektno orijentisana distribuirana arhitektura koja je dostigla status industrijskog standarda i postala široko prihvaćena u aplikacijama upravljanja telekomunikacionim mrežama. CORBA omogućava međusobnu komunikaciju aplikativnim komponentama pisanim različitim programskim jezicima, a koje se izvršavaju unutar različitih procesa bilo na istom ili različitim računarima.

CORBA umotava programski kod u paket koji sadrži informacije o tome šta je sadržaj paketa i kako se isti može izvršiti.

CORBA je okruženje koje omogućava razvoj i izvršavanje distribuiranih aplikacija, a to može biti potrebno (nužno) iz slijedećih razloga:

**----- OSTATAK TEKSTA NIJE PRIKAZAN. CEO RAD MOŽETE
PREUZETI NA SAJTU. -----**

www.maturskiradovi.net

MOŽETE NAS KONTAKTIRATI NA E-MAIL: maturskiradovi.net@gmail.com