



Princip rada CORBA tehnologije na primeru rada banke Principle of CORBA technology on the example of the bank

student: Ivan Panić, Prirodno-matematički fakultet u Novom Sadu,
ivanpanic@live.com

preduzeće/supervisor: TelventDMS, d.o.o., Novi Sad, Tijana Brcanski, dipl. ing.
tijana.brcanski@telventdms.com

mentor prakse: Prof. Miloš Racković, Prirodno-matematički fakultet u Novom
Sadu, ravkovic@dmi.uns.ac.rs



1. UVOD

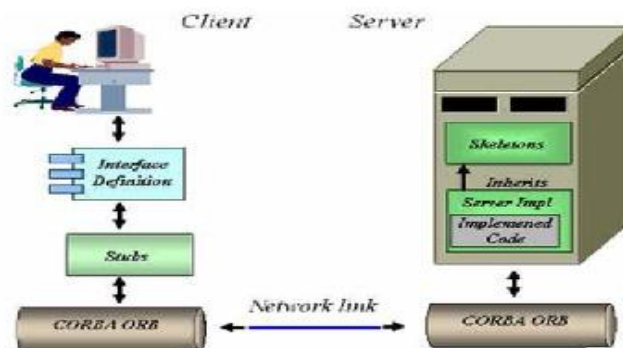
U okviru stručne prakse radio sam u firmi TelventDMS d.o.o. U firmi sam se upoznao sa CORBA tehnologijom pomoću omniORB biblioteke u programskom jeziku Java, okruženje Eclipse. Tokom boravka u firmi radio sam proširivanje i izmenu nekih postojećih biblioteka, a sa CORBA tehnologijom sam se upoznao detaljno pomoću primera banke koji ujedno i predstavlja moj rad za praksu.

2. CORBA – OSNOVNO

Common Object Request Broker Architecture ili skraćeno CORBA je standard definisan od strane Object Management Group (OMG) koji omogućava međusobnu komunikaciju aplikativnim komponentama pisanim različitim programskim jezicima, a koje se izvršavaju unutar različitih procesa bilo na istom ili različitim računarima [1]. Ova tehnologija omogućava da posebni delovi softvera pisani na različitim računarima, kao i u različitim programskim jezicima rade zajedno kao jedna aplikacija. CORBA umotava programski kod u paket koji sadrži informacije o tome šta je sadržaj paketa i kako se isti može izvršiti. CORBA koristi IDL (Interface Definition Language) za specifikaciju interfejsa koje će objekti predstaviti spoljašnjem svetu i specifikuje način mapiranja IDL-a u odgovarajući programski jezik (u ovom slučaju Java). Standardna mapiranja postoje za programeske jezike: Java, Ada, C, C++, Lisp, Ruby, Smalltalk, COBOL, PL/I i Python, a što se tiče ne standardnih mapiranja, tu spadaju: Perl, Visual Basic, Erlang i Tcl. [2]

3. CORBA – ARHITEKTURA

Sa klijentske strane aplikacija definiše referencu na distribuirani objekat koji u sebi ima „*stub*“ metodu.



Slika 1 – Klijent – server arhitektura (CORBA)

Ova metoda mehanizme povezivanja na ORB kojim se zahtev prosleđuje distribuiranom objektu. Na serverskoj strani postoji „*skeleton*“, metoda koja prevodi upit i parametre pristigle od klijenta i koja dalje poziva izvršavanje metode na distribuiranom objektu.

4. PRAKTIČNI DEO

Sam rad obuhvatio je princip rada jedne banke. Postoje strukture: *Bank*, *Employed*, *Account*, *Credit* i *Client*, metode: *PayBill()*, *GetClientWithMostCredits()*, *GetBankInformation()*, *GetAllBankIDs()* itd. sa klijentske strane, metode dodavanja struktura sa serverske strane, kao i „*callback*“ koji je vraćao da li je klijent otisao u minus, informaciju da li je klijent dodat i da li je račun promenjen (*OnAccountMinus()*, *OnClientAdd()*, *OnAccountChange()*). Akcenat rada je upoznavanje sa svim elementima CORBA tehnologije. Obuhvaćeni su još: izuzeci, „*callback*“, konstante, enumeracije, kao i tip „*Any[]*“ (niz koji u sebi može da sadrži bilo koju strukturu ili bilo koji tip podatka zajedno).

5. ZAKLJUČAK

CORBA predstavlja jednu od najboljih tehnologija za realizovanje klijent – server arhitekture. Njena osnovna prednost je univerzalnost. Ona definiše samo način na koji se vrši komunikacija između fizički odvojenih objekata. Sama implementacija se vrši na osnovu definisanih principa. Ovakav princip omogućava CORBA funkcionalnost na svim operativnim sistemima, kao i univerzalno korišćenje većine objektno orijentisanih jezika.[3]

LITERATURA

1. Zvonimir Vuković, Common Object Request Broker Architecture, 2007, <http://www.etfos.hr/upload/OBAVIJESTI/obavijesti_strucni/15764corba.pdf>
2. <http://en.wikipedia.org/wiki/Common_Object_Request_Broker_Architecture>.
3. <<http://java.sun.com/j2ee/corba/>>