Softversko inženjerstvo

Elektronski fakultet Niš

Arhitekturni modeli

Arhitekturni model (stil)

- Arhitekturni modeli (stilovi) predstavljaju projektne obrasce za arhitekturu SW-a.
- Oni definišu komponente i konektore koji čine arhitekturu sistema.
- Na ovaj način, arhitektura sistema se može predstaviti kao graf čiji su čvorovi sledeće komponente:
  - Procedure;
  - Moduli;
  - Procesi;
  - Alati;
  - Baze podataka;

Arhitekturni model (stil)

- Grane (potege) grafa čine konektori koji mogu biti:
  - Pozivi procedura;
  - Prenosi događaja (evenata);
  - Upiti baze podataka;
  - Protočne obrade.

Osnovni arhitekturni modeli

- Repository (Skladište)
- Pipe and Filter (Protočna obrada)
- OO model
- Client/Server model
- Slojeviti model (Layered)
- Event-driven model (Implicitno pozivanje)
- Control model

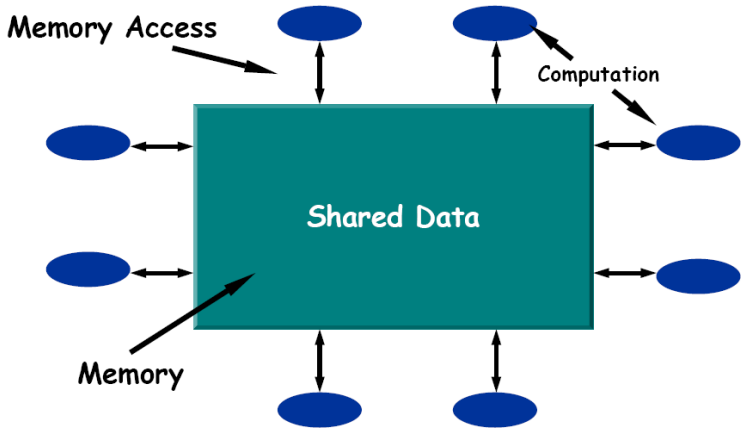
Repository model

- Ovaj model se koristi kod sistema kod kojih je neophodno deljenje velikih količina podataka.
- Tada se organizuje centralno skladište podataka kome pristupaju podsistemi.
- Komponente:
  - Centralna baza podataka.
  - Skup SW komponenti koje pristupaju centralnoj bazi.

Repository model

- Konektori:
  - Pozivi procedura.
  - Direktan pristup bazi podataka.

Repository model



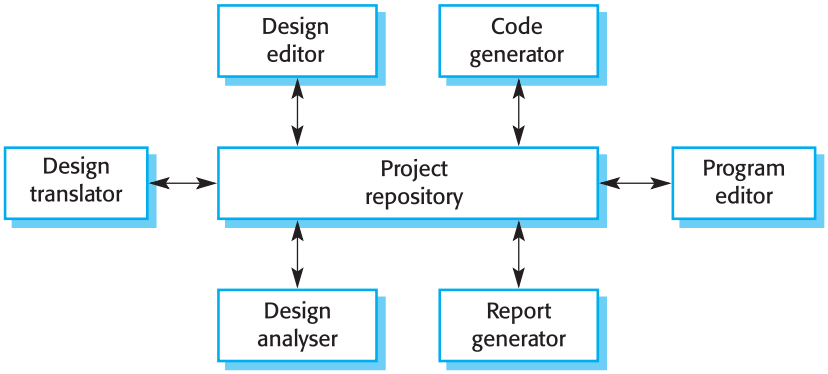
Repository model - karakteristike

- Prednosti
  - Efikasan način za deljenje velike količine podataka;
  - Podsistemi ne moraju da brinu o nekim aspektima upravljanja podacima kao što su backup, sigurnost,...
  - Model deljenja podataka je prikazan u obliku šeme skladišta podataka.
- Nedostaci
  - Podsistemi moraju da se slože oko skladišta podataka. Kompromis je neizbežan;
  - Evolucija podataka je skupa;
  - Otežana je distribucija podataka.

Repository model - primeri

- Informacioni sistemi
- Okruženja za razvoj SW-a
- Grafički editori
- Baze znanja u VI
- Sistemi za reverse engineering

Repository model - primer



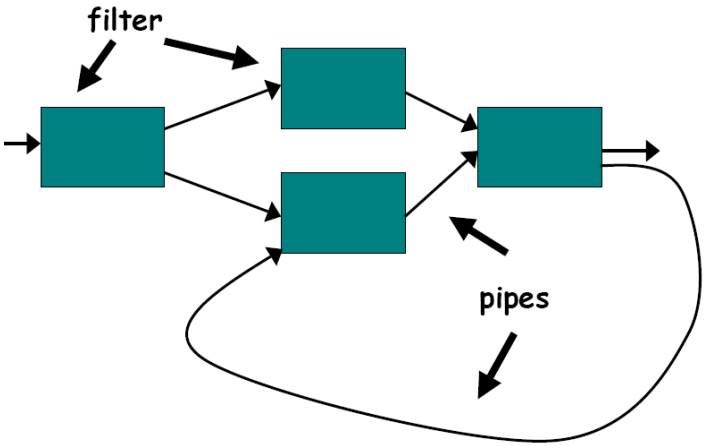
Pipe and Filter model (protočna obrada)

- Ovaj model se koristi kod sistema kod kojih je neophodno izvršiti unapred definisane serije nezavisnih obrada nad podacima.
- Komponente prihvataju tokove podataka na ulazu i generišu tokove podataka na izlazu.
- Komponente:
  - Komponente se često zovu i filtri koji vrše određene transformacije ulaznih podataka.

Pipe and Filter model

- Konektori:
  - Konektori se često zovu pipe (cevovod) jer povezuju tokove i filtre (izlaz jednog filtra vode na ulaz drugog filtra).

Pipe and Filter model



Pipe and Filter model - karakteristike

- Prednosti
  - Lako razumevanje ulazno-izlaznog ponašanja celokupnog sistema kao skupa individualnih ponašanja filtera u sistemu;
  - Lako je ponovno korišćenje filtera (reuse), jer je između dva filtra već definisan format razmene podataka.
  - Laka je izmena ili proširenje sistema (zamenom postojećih ili dodavanjem novih filtera).
  - Prirodno je podržano konkurentno izvršenje.
- Nedostaci
  - Nije najbolji izbor kod interaktivnih sistema zbog velikog broja transformacija;
  - Povećava kompleksnost sistema i smanjuje efikasnost;

Pipe and Filter model - primeri

- Unix shell skriptovi

```
cat file | grep Erroll | wc -l
```

- Tradicionalni kompajleri (prevodioci)

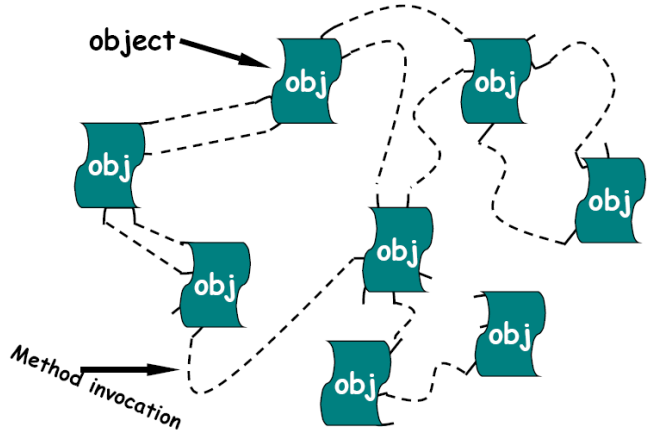
lexical analysis + parsing + semantic analysis + code generation

Pipe and Filter model - primer

OO model

- Ovaj model se koristi kod sistema kod kojih je neophodno izvršiti zaštitu i enkapsulaciju podataka.
- Podaci i pridružene operacije su enkapsulirane u apstraktne tipove podataka (klase i objekte).
- Komponente:
  - Objekti
- Konektori:
  - Metodi (servisi) objekata.

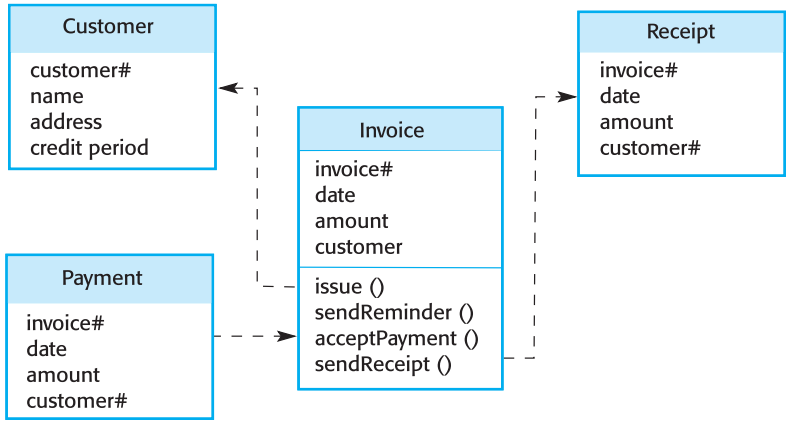
OO model



OO model - karakteristike

- Prednosti
  - Zbog skrivanja informacija od klijenata, moguće je promeniti implementaciju objekata bez uticaja na klijente;
  - Moguće je projektovanje sistema kao skupa autonomnih agenata.
  - Moguće je direktno mapiranje entiteta iz realnog sveta u objekte.
- Nedostaci
  - Neophodno je poznavanje indentiteta objekata. Ukoliko se izmeni identitet nekog objekta, moraju se izvršiti izmene i kod svih objekata koji ga pozivaju;
  - Mogućnost pojave “bočnih efekata”;

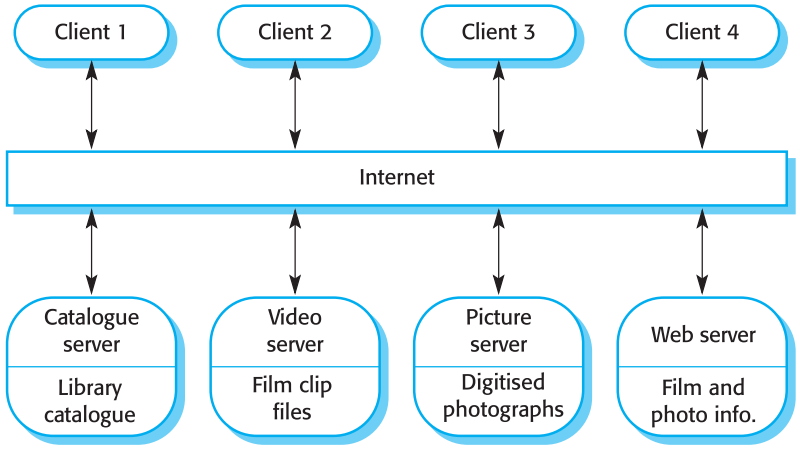
OO model - primer



Client-Server Model

- Ovaj model se koristi kod distribuiranih sistema.
- Sastoji se od skupa stand-alone servera koji obezbeđuju specifične servise (štampa, Web, baza podataka,...), skupa klijenata koji pozivaju te servise i mreže koja omogućuje udaljeni pristup.
- Komponente:
  - Serveri, klijenti.
- Konektori:
  - Mreža, servisi servera.

Client-Server Model - primer



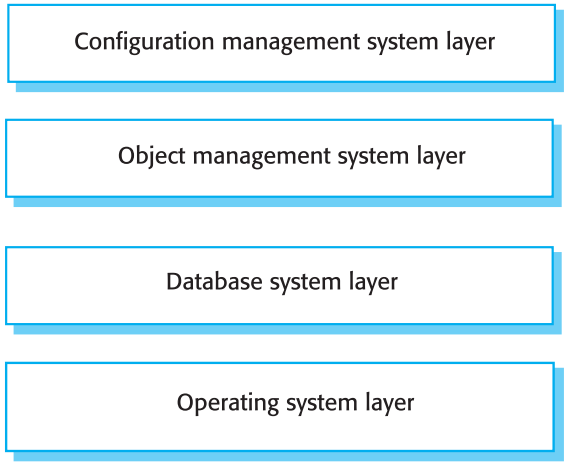
Client-Server model - karakteristike

- Prednosti
  - Efikasno korišćenje mrežnih sistema;
  - Omogućava korišćenje slabijeg HW-a za klijente, obzirom da server odrađuje većinu posla.
  - Lako dodavanje novih servera i upgrade postojećih.
- Nedostaci
  - Neefikasna razmena podataka između klijenata (moraju da idu preko servera);
  - Redundantnost podataka.
  - Ne postoji centralni registar imena servera i servisa; Nije lako otkriti koji serveri i servisi su na raspolaganju.

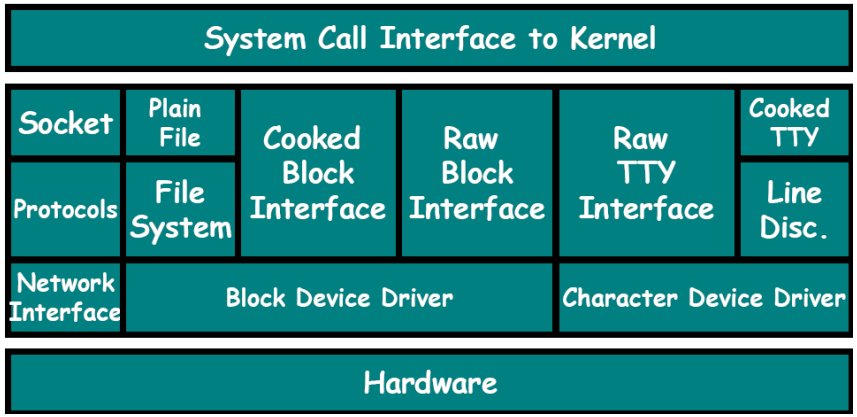
Slojeviti (Layered) Model

- Ovaj model se koristi kod modeliranja interfejsa među podsistemima.
- Sistem se organizuje u skup slojeva (layera) od kojih svaki obezbeđuje jedan skup funkcionalnosti sloju iznad i služi kao klijent sloju ispod.
- Omogućava inkrementalni razvoj podkomponenti u različitim slojevima.
- Komponente:
  - Slojevi.
- Konektori:
  - Interfejsi.

Layered model – primer sistema za kontrolu verzija



Layered model – primer OS Unix



Layered model - karakteristike

- Prednosti
  - Promena interfejsa jednog sloja može da utiče na maksimalno još dva sloja;
  - Laka zamena jednog sloja drugim ukoliko su im interfejsi identični.
  - Baziran je na visokom nivou apstrakcije.
- Nedostaci
  - Ne mogu svi sistemi da se lako organizuju po ovom modelu;

Event driven model (Implicitno pozivanje)

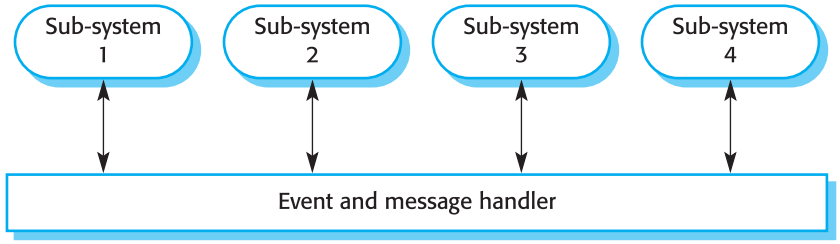
- Ovaj model se koristi kod sistema koji su upravljani eksterno generisanim događajima (events).
- Postoje dve osnovne grupe ovih modela:
  - Broadcast modeli
  - Interrupt-driven modeli
- Komponente:
  - Komponente i podsistemi koji generišu ili obrađuju evente.
- Konektori:
  - Broadcast sistem i event procedure.

Broadcast modeli

- Kod ove grupe modela, generisani događaj (event) se prosleđuje svim komponentama i podsistemima u sistemu. Svaka komponenta koja upravlja generisanim događajem može da obradi događaj.
- Efikasni su kod integracije podsistema koji se nalaze na različitim računarima u mreži.
- Podsistemi neznaju da li će i kada eventi biti obrađeni.
- Podsistemi se registruju za određene događaje i kada se oni generišu, upravljanje se prenosi na podsistem koji upravlja tim događajem.

Elektronski fakultet u Nišu

Broadcast model – primer



Elektronski fakultet u Nišu

Broadcast model – primer razvojnog okruženja

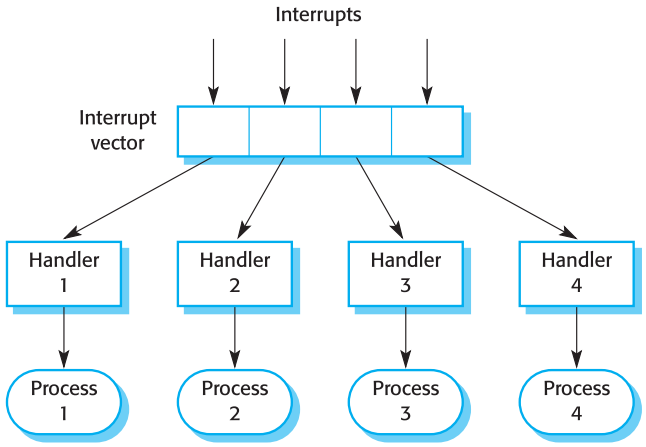
- Ovaj model se često koristi kod razvojnih okruženja za integraciju alata:
  - Debager se zaustavi na prekidnoj tački i generiše događaj da je to uradio.
  - Editor odgovara na taj događaj tako što skroluje sadržaj koda na liniju gde je postavljena prekidna tačka.
  - ...

Elektronski fakultet u Nišu

Interrupt-driven modeli (modeli upravljani prekidima)

- Koriste se kod sistema za rad u realnom vremenu gde je osnovna stvar brzi odgovor sistema na neki događaj.
- Obezbeđuju brzu reakciju sistema na događaje, ali su komplikovani za realizaciju i pogotovo za testiranje i validaciju sistema.

Interrupt-driven model – primer



Event driven model - karakteristike

- Prednosti
  - Podrška višestrukom korišćenju SW-a (reuse);
  - Laka evolucija sistema.
  - Lako uvođenje nove komponente u sistem (jednostavno se registruje za neki event).
- Nedostaci
  - Kada komponenta generiše događaj ona ne može da zna da li će neka komponenta da odgovori na njega i kada će obrada događaja biti završena;

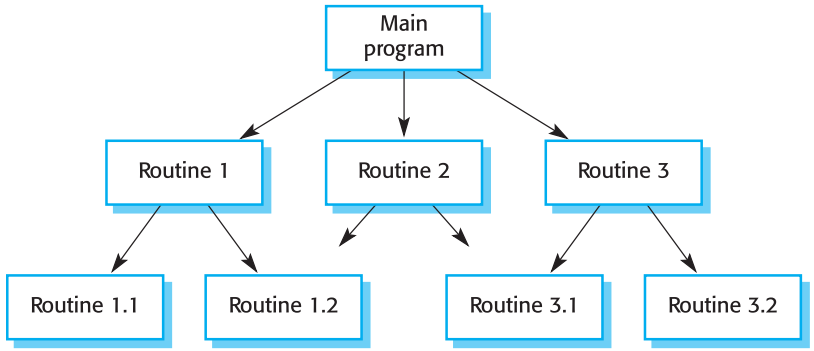
Control model

- Koristi se kod sistema gde je potrebna centralizovana kontrola.
- Kontrolni podsistem upravlja tokom informacija između ostalih podsistema.
- Postoje četiri osnovne grupe ovih modela:
  - Call-return modeli
  - Manager modeli
  - Feed-back modeli
  - Open-loop modeli
- Komponente:
  - Kontrolni algoritam i podsistemi.
- Konektori:
  - Relacije između tokova podataka.

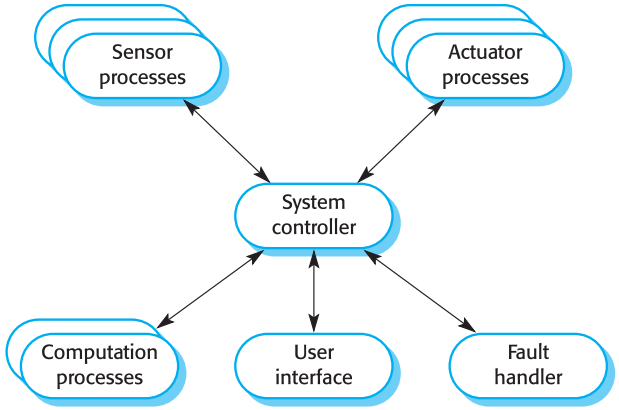
Call-return modeli

- Kod ove grupe modela, kontrola kreće od vršnih podsistema i proteže se naniže (top-down pristup).
- Pogodni su za sekvencijalne sisteme.

Call-return model – primer



Call-return model – primer sistema za rad u realnom vremenu

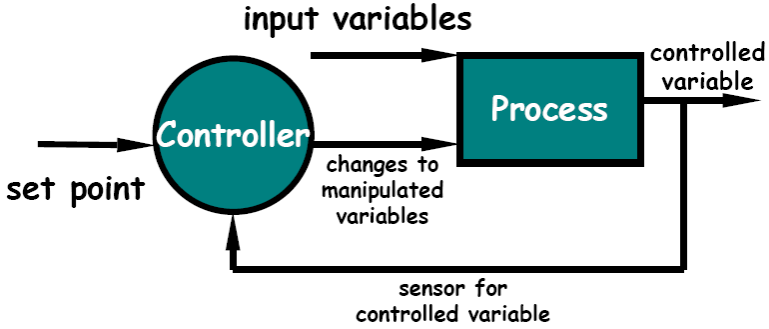


Manager modeli

- Ova grupa modela se primenjuje kod konkurentnih sistema.
- Jedna sistemska komponenta određuje početak, zaustavljanje i koordinaciju rada svih procesa u sistemu.

Feed-back modeli

- Kod ove grupe modela, neke promenljive sistema se kontrolišu i njihove vrednosti se koriste za podešavanje sistema.



Open-loop modeli

- Kod ove grupe modela, neke promenljive sistema se kontrolišu ali se njihove vrednosti ne koriste za podešavanje sistema.