

Ohjelmistoarkkitehtuurit

Arkkitehtonisista tyyleistä ja ratkaisumalleista jatkuu

13.9.2012

1

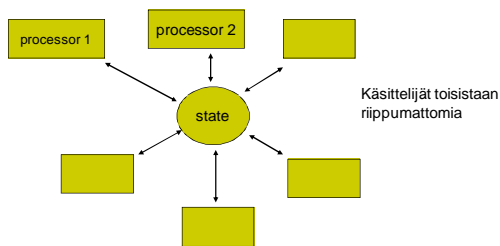
Jaettuun tietovarastoon perustuva arkkitehtuuri

- **Data centered - Shared state - Repository**
- Joukko komponentteja ylläpitää yhteistä tilaa tietovarastossa
 - Erilaisia muunnelmia sen mukaan, kuinka aktiivinen rooli tietovarastolla on
- Kahdenlaisia komponentteja
 - Keskitetty tietorakenne: tietovarasto
 - Asiakaskomponentit: hakevat tietoa tietovarastosta ja muokkaavat sitä
- Järjestelmän kontrolli määräytyy tietovaraston tilan mukaan

13.9.2012

2

Jaettuun tietovarastoon perustuva arkkitehtuuri



13.9.2012

3

Jaettuun tietovarastoon perustuva arkkitehtuuri

- Vuorovaikutus käsittelijöiden välillä tapahtuu ainoastaan tietovaraston kautta
- Käsittelijöiden tietovarastoon tekemät muutokset johtavat vaiheittain haluttuun lopputulokseen
- Käsittelijän aktivointi
 - Käsittelijät voivat pollata tietovarastoa tutkiakseen onko tila sellainen, mistä käsittelijä pystyy jatkamaan toimintaa
 - *jos on, käsittelijä tuottaa tuloksia, jotka kirjaa tietovarastoon*
 - Tietovarasto voi aktivoida käsittelijän jonkin säännön perusteella, esimerkiksi sisällön muuttumisen perusteella
 - *vrt tietokantatriggerit*

13.9.2012

4

Jaettuun tietovarastoon perustuva arkkitehtuuri

- Käsittelijät voivat toimia rinnakkain
 - Edellyttää samanaikaisuuden hallintaa: tiedon lukitus → miten estetään yhtä käsittelijää varaamasta koko tietovarastoa itselleen (liian pitkäksi aikaa)?

13.9.2012

5

Tulkkiarkkitehtuuri

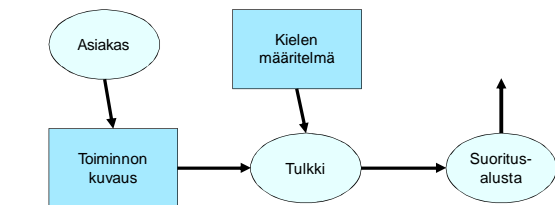
- Toimintatapa selvillä vasta suoritusaikana:
 - Järjestelmä tarjoaa joukon primitiivipalveluita
 - Vasta suoritusaikana tiedetään, kuinka näitä yhdistelemällä saadaan aikaan haluttu toiminnallisuus
 - *Tarve antaa järjestelmälle syötteenä toiminnallisia kuvauksia → tulkkipohjaiset arkkitehtuurit*
- Toinen käyttötilanne
 - Halutaan esittää sovelluksen toiminnallisuus abstraktina toiminnallisuutena, jotta sovelluksesta tulisi helposti siirrettävä eri laitealustoille
 - eri laitealustoja varten toteutetaan **virtuaalikoneet**, jotka kaikki tulkitsevat samaa abstraktia toiminnallisuuden kuvausta (esim. Java bytecode)

13.9.2012

6

Tulkkiarkkitehtuuri

- Perusajatus: tulkki lukee askeleittain (käsky käskyttä) toiminnallista kuvausta ja kutsuu vastaavia alustan tarjoamia palveluita



13.9.2012

7

Tulkkiarkkitehtuuri

- Liikkuva koodi (mobile code)
- Ohjelmakoodi siirretään toiseen paikkaan suoritettavaksi – suoritus usein tulkitsevaa esim. JavaScript, Flash,...
- Ladattava koodi (Code on demand)
 - koodi tuodaan tilan ja resurssien luo
- Lähetettävä koodi (Remote evaluation)
 - koodi viedään resurssien luo - grid computing
- Lähetettävä koodi + tila (Mobile agent)
- Edellyttävät suoritusalustaa
- Turvallisuusriskit

13.9.2012

8

Viestinvälitysarkkitehtuurit

- Implicit invocation
- Komponentit kommunikoivat lähettämällä toisilleen viestejä
- Viesti voidaan lähettää tietyille kohteelle tai yleislähetystenä kaikille tunnetuille kohteille tai kaikille kiinnostuneille kohteille
- Viestintä voi olla
 - synkronista (lähettäjä jää odotamaan) tai
 - asynkronista (lähettäjä jatkaa toimintaansa)

13.9.2012

9

Viestinvälitysarkkitehtuurit

- Julkaisija – Tilaaaja (Publish – Subscribe)=
- Tuottaja – Kuluttaja (Supplier- Consumer)
- Yksinkertaisimmillaan julkaisija pitää kirjaa vastaanottajista ja tietosisällön muuttuessa välittää tiedon tilaajille proseduurikutsun välityksellä.
- Tilaaaja rekisteröityy vastaanottajaksi ja toimittaa rekisteröinnin yhteydessä vastaanottorajapinnan (callback)
- Yllä oleva toimii ohjelman sisäisesti
- Verkkoratkaisussa tarvitaan kommunikointiprotokolla ja välittäjä huolehtimaan viestinvälityksestä.

13.9.2012

10

Viestinvälitysarkkitehtuurit

- Välittäjä tietää vastaanottajat (rekisteröityminen tai konfigurointi)
- Julkaisija toimittaa viestin välittäjälle.
- Välittäjä toimittaa viestin edelleen siitä kiinnostuneille tilaajille.
- Tilaaaja käsittelee viestin.

13.9.2012

11

Viestinvälitysarkkitehtuurit

- Arkkitehtuurissa määriteltävä seuraavat asiat:
 - Keskenään kommunikoivat komponentit
 - Viestit, joiden avulla kommunikointi tapahtuu ilman, että lähettäjä tuntee vastaanottajan (tai vastaanottaja lähettäjän) sijaintia
 - Operaatiot, joilla komponentit reagoivat viesteihin (komponenttien ymmärtämä kieli)
 - Säännöt, joiden avulla komponentit ja viestit rekisteröidään järjestelmälle

13.9.2012

12

Viestinvälitysarkkitehtuurit

- Säännöt, joiden perusteella välittäjä tietää, mille komponentille viesti on lähetettävä
 - Viestin vastaanottajan selvittäminen:**
 - yleisviesti (multiple dispatch; viesti kaikille, jotkut käsittelevät, toiset eivät)
 - etu: vastaanottajan ei tarvitse etukäteen kertoa, mitä viestejä se haluaa vastaanottaa
 - komponentit kertovat rekisteröinnin yhteydessä, mitä viestejä ne haluavat vastaanottaa (tai keneltä)
- Rinnakkaisuus: missä määrin välittäjä ja komponentit toimivat rinnakkain
 - viestinvälittäjillä ja vastaanottajilla viestijonot (välittäjän laatikko, vastaanottajan laatikko)

13.9.2012

13

Viestinvälitysarkkitehtuurit

- Tapahtumapohjaiset arkkitehtuurit (event-based, event driven)
- Komponentit kommunikoivat aiheuttamalla tapahtumia (event)
 - Tapahtumat voivat välittyä kaikille muille komponenteille, joista jotkut käsittelevät ja toiset eivät noteeraa
 - Tapahtumien lähetyksen voidaan myös rajoittaa julkaisija – tilaaja rakenteen tapaan – tapahtuma lähetetään vain tapahtuman tyyppistä kiinnostuneille
 - komponentteja ei ole kuitenkaan lokeroitu joko julkaisijoiksi tai tilaajiksi vaan komponentti voi toimia kumpunkin

13.9.2012

14

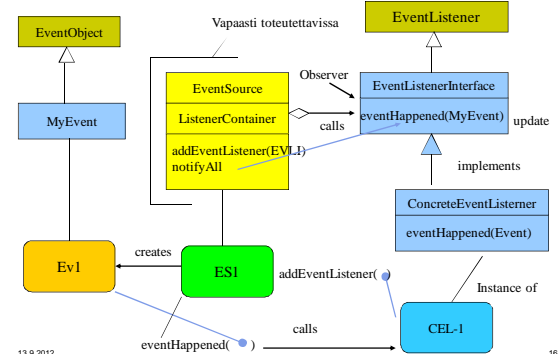
Viestinvälitysarkkitehtuurit

- Tapahtuma
 - Tilanne, joka voi sattua ohjelman suorituksen aikana
 - Edellyttää reagoitua joiltain järjestelmän osilta
 - Lähde** = tapahtuman synnyttävä komponentti
 - Tarkkailija** = tapahtumaan reagoiva komponentti
- Lähde lähettää tapahtumailmoituksen sitä tarkkailemaan rekisteröityneelle tarkkailijalle
- Lähde tietää dynaamisesti tarkkailijoidensa olemassaolon, mutta ei niiden tarkkaa tyyppiä
- Ilmoituksen lähetyksen voidaan hoitaa proseduurikutsuna tai sanomanvälityksenä (event bus, message bus)
 - proseduurikutsu synkroninen
 - Sanomajono asynkroninen/synkroninen

13.9.2012

15

Javan tapahtumamalli

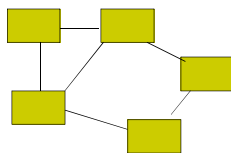


13.9.2012

16

Vertaisverkkoarkkitehtuuri

- peer-to-peer
- verkko, jonka solmut voivat toimia sekä asiakkaina että palvelimina



puhtaassa vertaisverkossa kaikki solmut tasavertaisia

13.9.2012

17

Vertaisverkkoarkkitehtuuri

- Palvelupyynnöt etenee verkossa kunnes löytyy solmu, joka kykenee täyttämään pyynnön
- Rakenteettomassa verkossa pyynnön eteneminen sokeaa
- Rakenteellisissa solmut jakavat rakennetietoja palveluista ja tätä käytetään ohjaamaan etenemistä
- Edellyttää verkkoprotokolla

13.9.2012

18