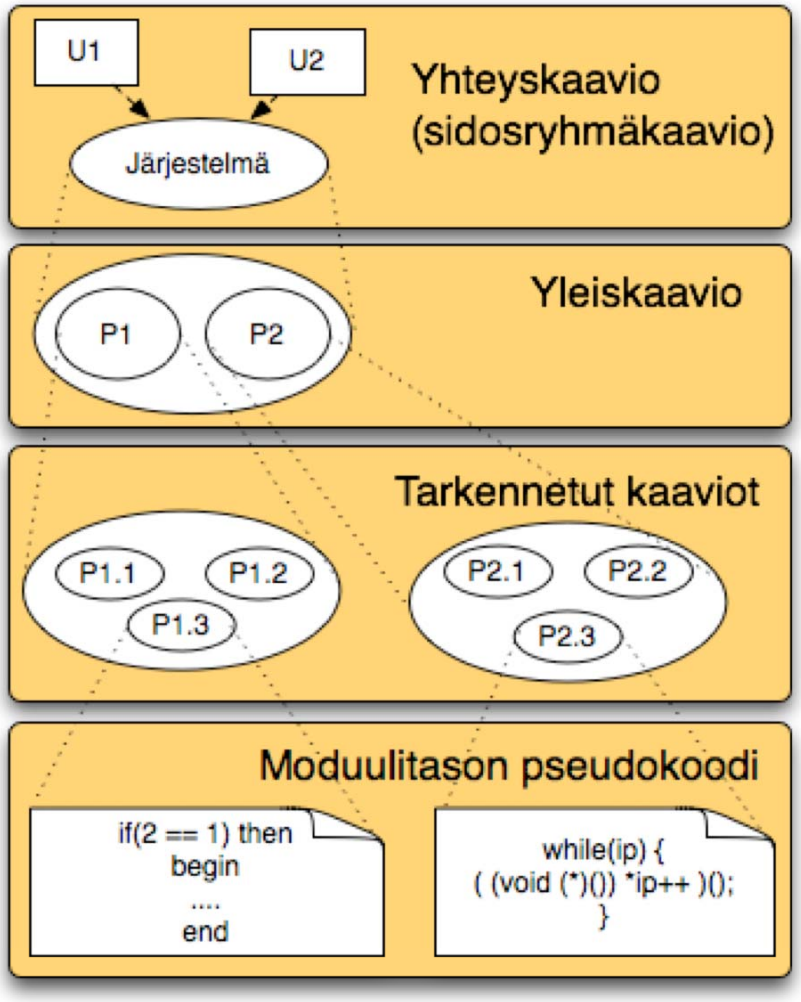
582104 – Ohjelmistojen mallintaminen
Unified Modeling Language (UML)

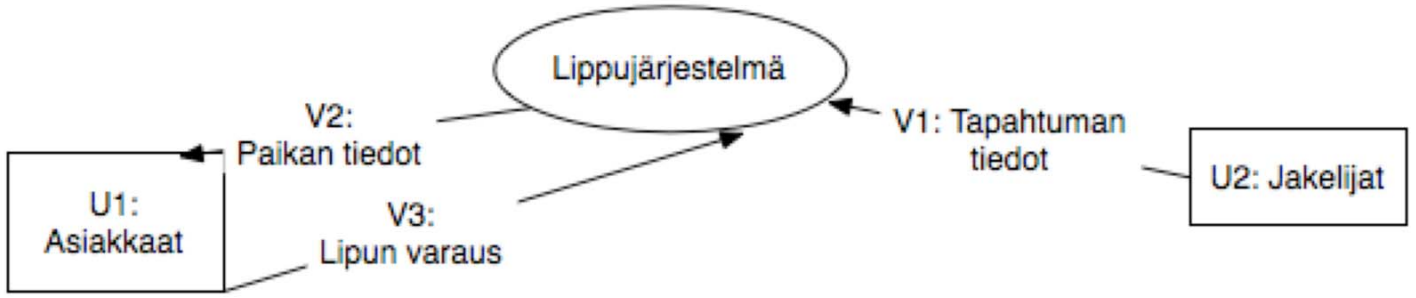
Tietovuokaaviot

- Data flow diagrams, DFD
- Historiallisesti käytetyin kuvaustekniikka
- Järjestelmän toiminnallinen ositus
- Epäyhteensopivuus oliomallinnuksen kanssa aiheuttanut suosion hiipumisen



Yhteys- eli sidosryhmäkaavio

- Rajaa järjestelmän suhteessa toimintaympäristöön
- Kaavioelementit
  - Yksi koko järjestelmää kuvaava prosessisymboli
  - Joukko ulkoisia tahoja (edustavat mm. järjestelmän sidosryhmiä ja muita, ulkoisia tietojärjestelmiä)
  - Joukko tietovuosymboleita em. välillä

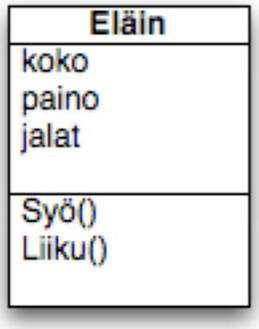


Olioperustaisuus

- *Olio* toimii mallinnuksen perusyksikkönä eri abstraktiotasoilla
  - Järjestelmän rajaus, suunnittelu, ohjelmointi, suoritus..
- Järjestelmä koostuu joukosta *olioita* (object), jotka yhdessä tuottavat järjestelmän palvelut
- Olio on *kokonaisuus* (entity), joka suorittaa omaan tietosisältöönsä perustuvia palveluita
- Olio kapseloi omat tietonsa tarjoamiensa palveluiden kautta käytettäviksi
- Oliolla on *identiteetti* (identity), joka pysyy vaikka olion *tila* (state) tai *käyttäytyminen* (behavior) muuttuisivat

Luokka

- Samanrakenteiset oliot kuuluvat samaan *luokkaan* (class)
- Eli ovat ko. luokan *ilmentymiä* (instance)



- Pitäisikö määritellä luokat kirahvi ja leijona ?

Luokka ja olio

```
public class Eläin {
    int eläinNumero;
    String laji;
    Color väri;
    float paino;

    public Eläin(int numero, String laji, Color väri, float paino) { …. }
    public float getPaino() { … }
}
```

Eläin leoLeijona = new Eläin(….);

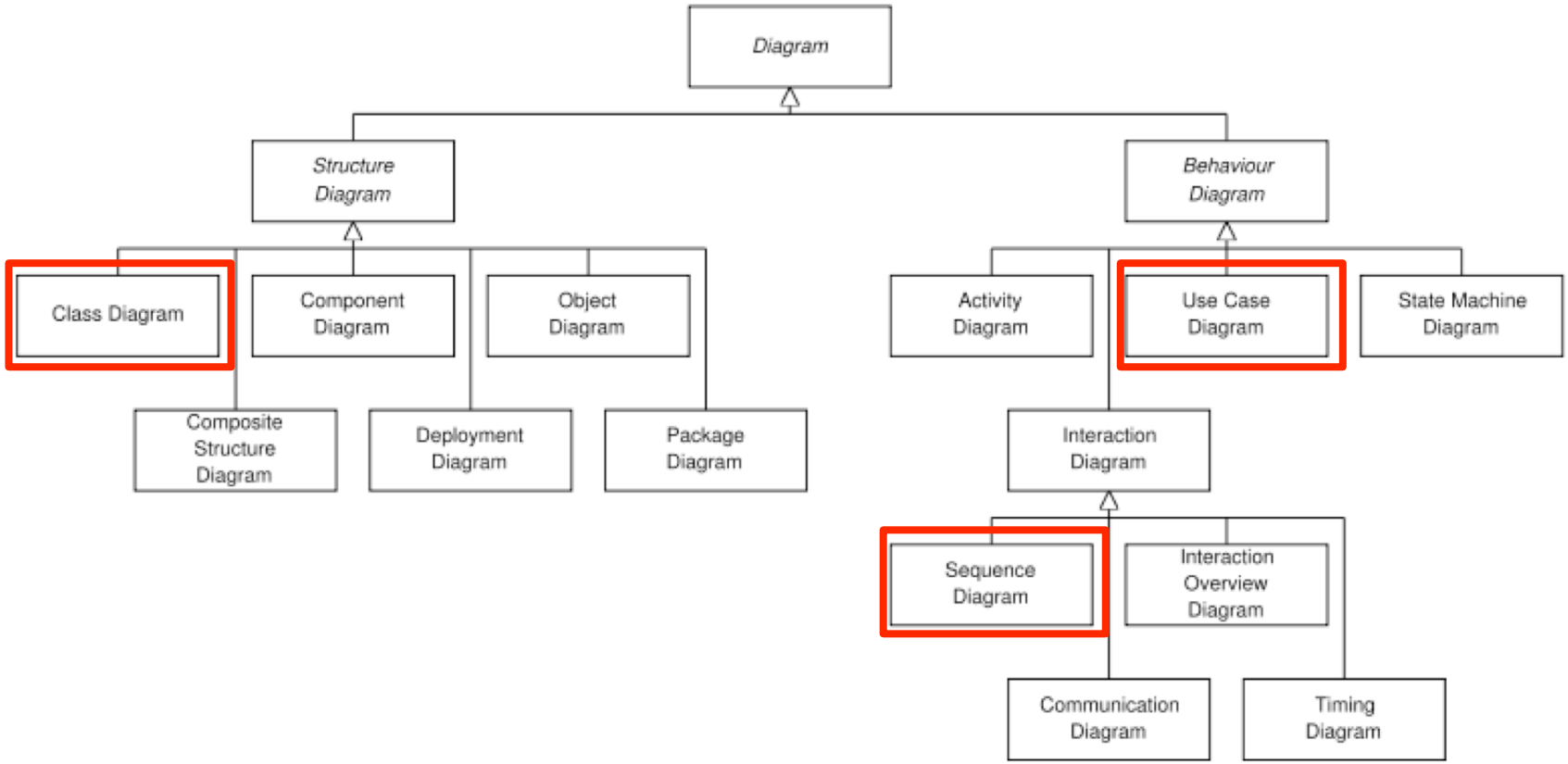
Eläin kiraKirahvi = new Eläin(…..);

leoLeijona.syö(kiraKirahvi);

Unified Modeling Language (UML)

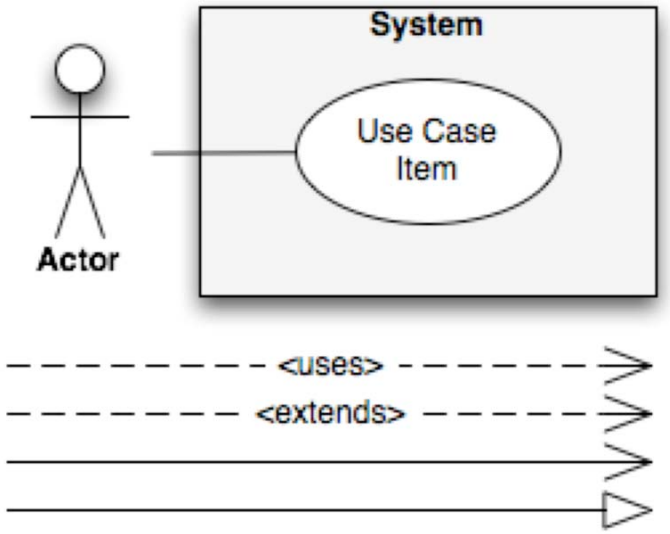
- ”… ohjelmistojen, liiketoiminnan ja muiden järjestelmien spesifiointiin, visualisointiin, rakentamiseen ja dokumentointiin tarkoitettu kieli.” [UML Specification 2003]
- Alun perin yhdistelmä vanhemmista oliotekniikoista
  - OMT (Rumbaugh) + Booch + OOSE (Jacobson)
- Kehitetty Rational Softwaren toimesta vuodesta 1996 lähtien
- Nykyisin teollisuusstandardi
- Useiden CASE-välineiden valinta
- Sisältää kymmenkunta kaaviotyyppiä

UML:n kaaviotyypit

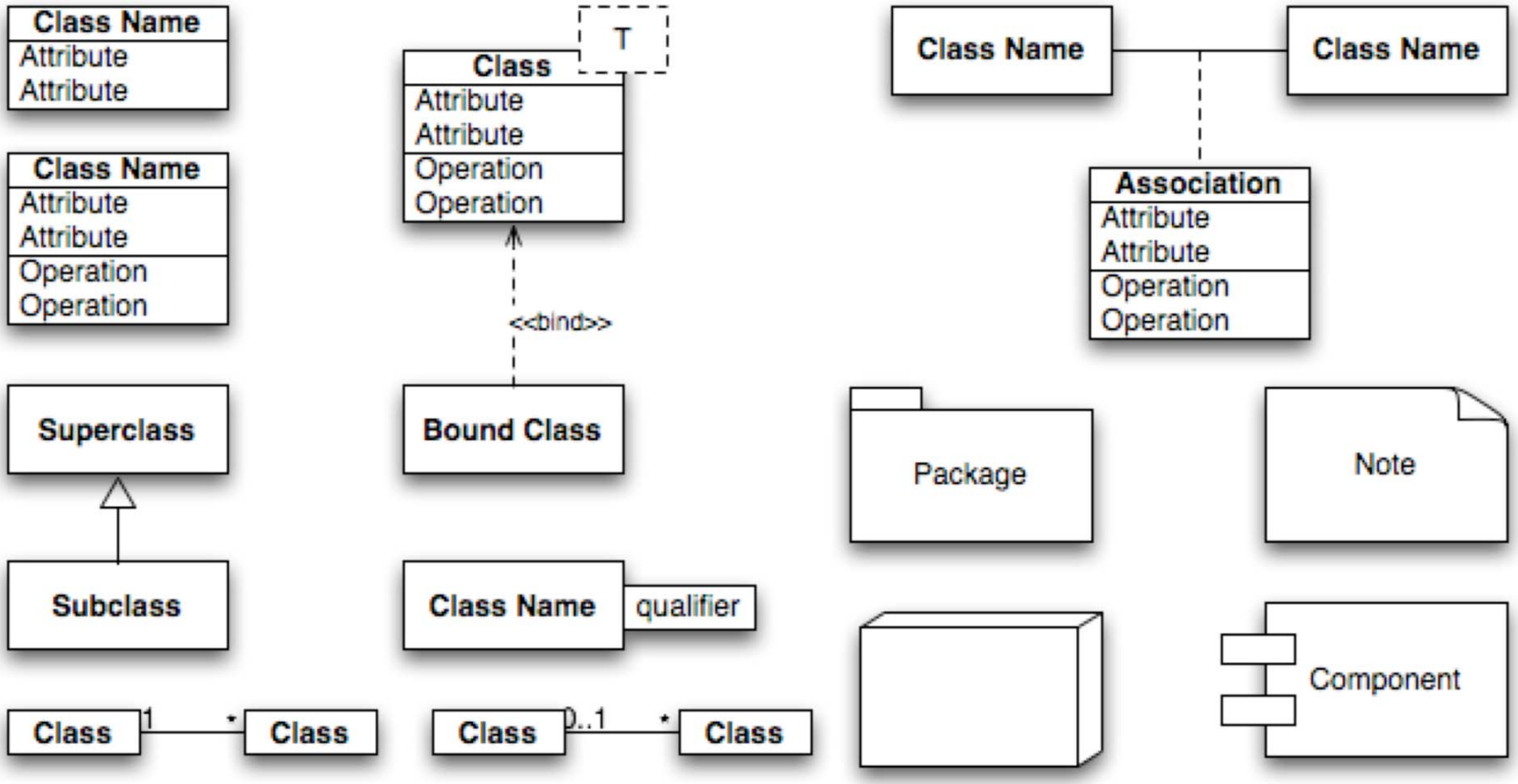


[Wikipedia]

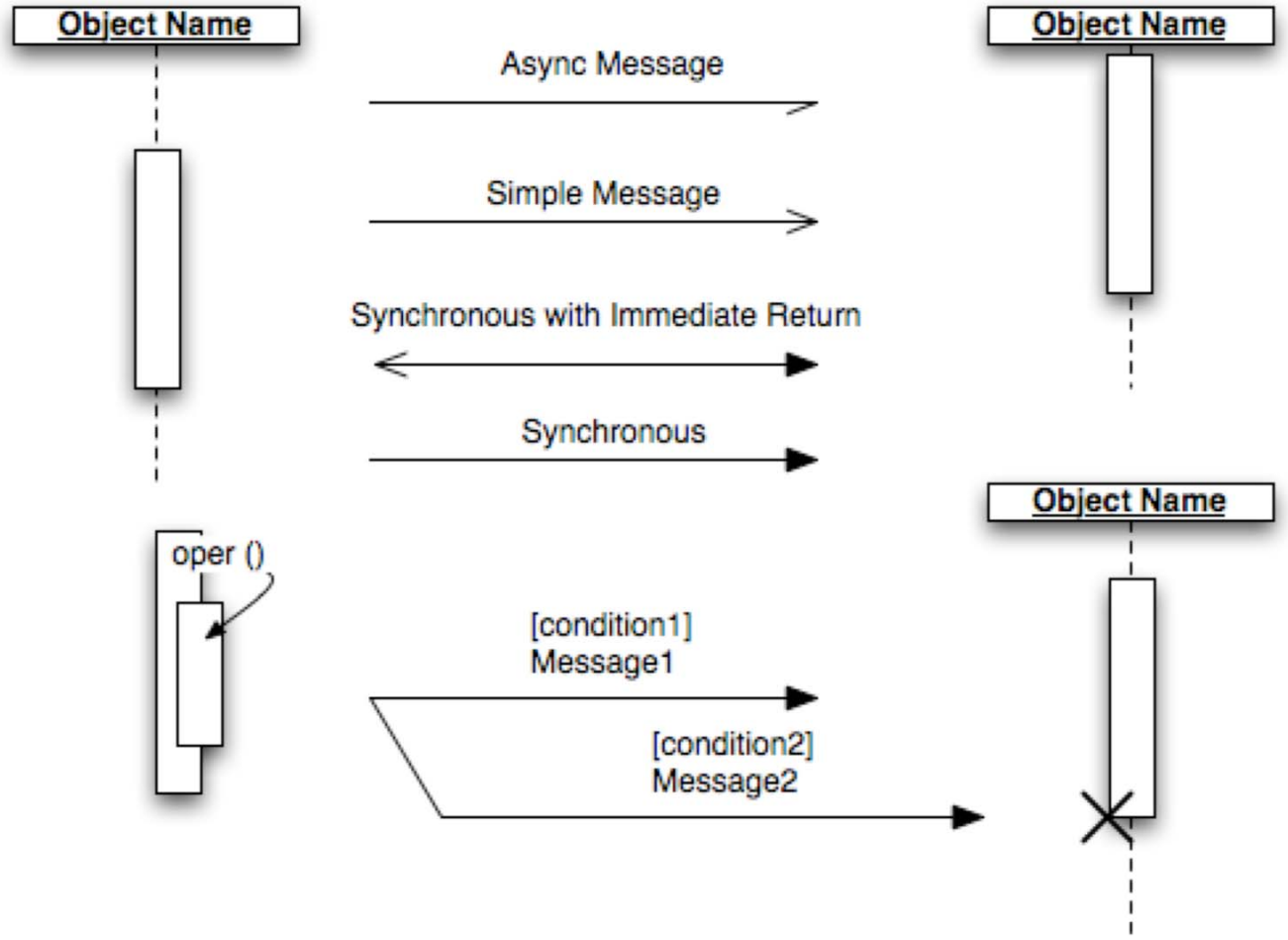
OmniGraffle-työkalun kategoria *UML Use Case*



OmniGraffle-työkalun kategoria *UML General*



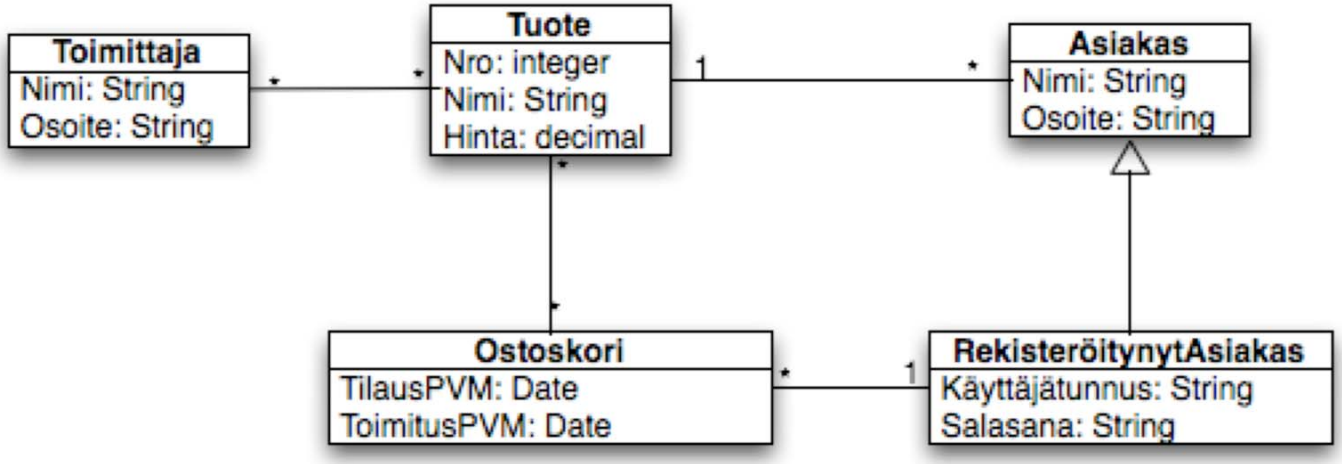
OmniGraffle-työkalun kategoria
UML Sequence



UML:n malli- ja kaaviotyypit: rakennemalli

- Ohjelman staattinen rakenne
- Kaaviotyypit:
  - *Luokkakaavio* ja oliokaavio
    - Järjestelmän tietosisältö ja käytettävissä olevat luokkien tarjoamat palvelut
    - Käytössä määrittely- ja suunnittelutasolla
  - Toteutuskaaviot eli komponentti- ja sijoittelukaavio (component diagram, deployment diagram)
    - Ohjelmiston koostuminen komponenteista ja niiden suoritusaikainen sijoittuminen

Esimerkki: Ostoskori



UML:n malli- ja kaaviotyypit: käyttäytymismalli

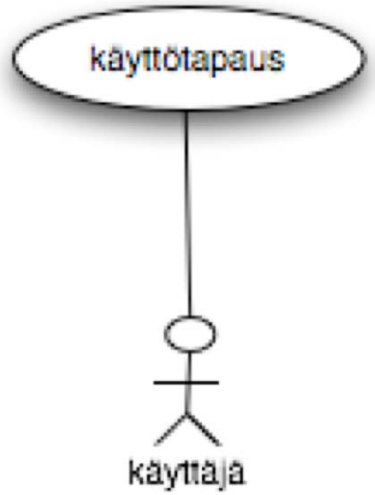
- Järjestelmän palvelut ja niiden toteuttaminen
- Kaaviotyypit:
  - *Käyttötapauskaavio* (use case diagram)
    - Mitä järjestelmällä tehdään
    - Tekstuaaliset kuvaukset olennaisia
  - Vuorovaikutuskaaviot eli *sekvenssi*- (sequence) ja *kommunikointikaavio* (communication diagram
    - Palveluiden toteuttaminen olioiden välisenä yhteistyönä
    - Sekvenssikaavio korostaa palveluiden käyttämistä aikajärjestyksessä
    - Kommunikointikaavio korostaa olioiden kytkentöjä

Käyttötapaukset

- Keskeisin tekniikka järjestelmän ulkoisesti havaittavien ominaisuuksien ja käyttäytymisen mallintamiseen määrittelytasolla UML:ssä
- UML määrittelee käyttötapauskaavion, mutta kaaviotekniikkaa olennaisempaa on kunkin käyttötapauksen tekstuaalinen kuvaus
- Käyttötapauksia voi kuvata eri tarkkuustasolla, mutta ne eivät muodosta varsinaista hierarkkista rakennetta

Käyttötapaus ja käyttäjä

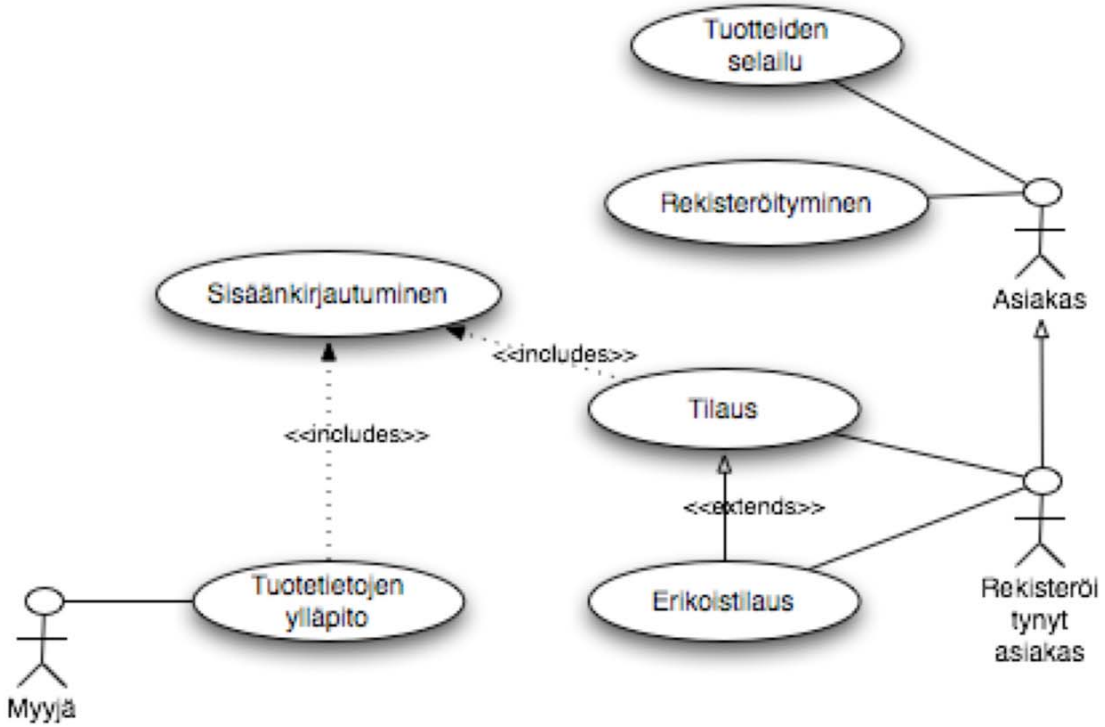
- **Käyttötapaus (use case)**
  - **Looginen tavoitteellinen tehtäväkokonaisuus, jolla on lähtökohta ja lopputulos**
  - **Usein järjestelmälle asetettava toiminnallinen vaatimus**
- **Käyttäjä (actor)**
  - **Rooli, jota järjestelmään liittyvä taho esittää**
  - **Usein ihminen, mutta voi olla myös ulkoinen järjestelmä**
  - **Toimii vuorovaikutteisesti järjestelmän kanssa**



Käyttötapauksen tekstikuvaus

- Sisältää tyypillisesti ainakin
  - Käyttötapaukseen liittyvät käyttäjät
  - Käyttötapauksen kulku askelittain
- Usein myös:
  - Annettavat syötteet ja saatavat tulosteet / tulokset
  - Säännöt, vaatimukset, määrät
  - Suhteet muihin tapauksiin, erikois- ja poikkeustilanteet

Käyttötapausesimerkki: ostoskori

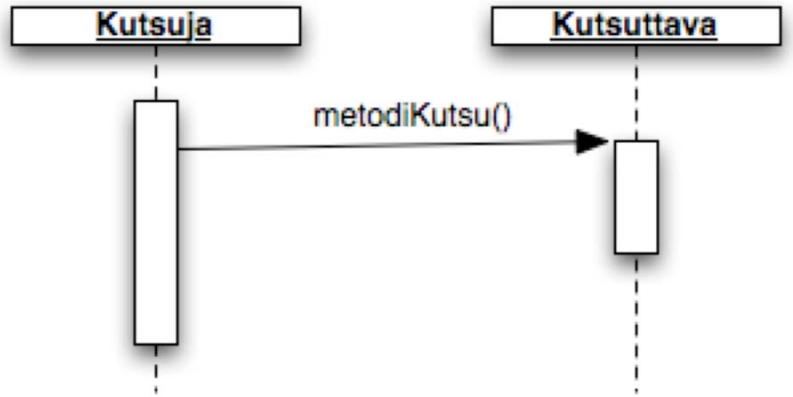


UML: vuorovaikutuskaaviot: sekvenssikaavio

- **Keskeisin tekniikka käyttäytymisen mallintamiseen suunnittelutasolla UML:ssä**
- **Visualisoi olioiden välisen viestinvälityksen**
  - ***Viesti*** **(message) kuvataan nuolena oliosta toiseen**
  - ***Kaaviossa aika*** **kulkee ylhäältä alas, viestit merkitään aikajärjestykseen**
  - **Viesti *aktivoi* metodin (eli palvelun tai operaation) suorituksen vastaanottavassa oliossa**
  - ***Aktivaatio*** **(activation) kuvataan suorakaiteena olion *elämänviivassa* (lifeline)**

Tärkeimmät sekvenssikaavion elementit

- ***Kutsuja*-olio kutsuu *Kutsuttava*-olion metodia nimeltä *metodiKutsu()***
- ***Paksunnettu pystypalkki kuvaa aktivaatiota***
- ***Horisonaaliset viivat kuvaavat viestien kulkua***



UML: malli- ja kaaviotyypit: tilanmuutosmalli (state change model)

- Järjestelmän tilan mahdolliset muutokset
  - Aktiviteettikaavio (activity diagram)
    - Kontrollin kulku prosessissa tai olion operaatiossa
    - Sisältää ehtoja ja haaraumia
  - Tilakaavio (statechart diagram)
    - Olion tilamuutokset esitetään tilakoneena
    - Tapahtumat (event) laukaisevat tilasiirtymiä (state transition) ja
    - Tuottavat toimintoja (actions)