

Ohjelmistotuotanto

---

Määrittelytekniikat 1

1

Järjestelmän rajaus

- Sidosryhmät (ulkopuoliset tahot)
  - ihmiset (käyttäjät), organisaatiot, toiset järjestelmät
  - ulkopuolella, mutta tekemisissä kehitettävän ohjelmiston kanssa
- Sidosryhmien väliset liittymät
  - tiedot tai palvelut
- Perinteinen kuvaustapa yhteyskaavio (context diagram)
  - kuvaa tiedon kulkua järjestelmän ja sidosryhmien välillä

© Harri Laine, Jukka Paakki 2

Järjestelmän rajaus

Tiedon kulkuun perustuva yhteyskaavio

© Harri Laine, Jukka Paakki 3

Järjestelmän rajaus

Palvelujen käyttöön perustuva yhteyskaavio

© Harri Laine, Jukka Paakki 4

Järjestelmän rajaus

Palvelujen käyttöön perustuva yhteyskaavio vapaamuotoisemmin

© Harri Laine, Jukka Paakki 5

1. Toiminnallisuuden kuvaus

- Käyttötapaukset (use case) - palvelukokonaisuudet ja niiden käyttö
  - Esimerkiksi UML:n käyttötapauskaaviot
- Tietovirrät (data flow) - tiedolle tehtävät muokkaustoimenpiteet ja tiedon kulku toimenpiteiden välillä
  - perinteinen kuvaustekniikka
- Tapahtumareaktiot (event processing) - reagointi ympäristön tapahtumiin

© Harri Laine, Jukka Paakki 6

### Käyttötapaukset

- Oliopohjainen lähestymistapa
- Alkuperä: Ivar Jacobsson
- Järjestelmä on olio. Oliolla on palveluja. Käyttötapaukset kuvaavat järjestelmän palveluja.
- Käsitelty JSS-kurssilla (Informaatiojärjestelmät) k97 alkaen

© Harri Laine, Jukka Paakki 7

### Käyttötapaukset

- Käyttötapaus
  - tavoitteellinen toimintokokonaisuus
  - mitä järjestelmällä voi tehdä
  - yleensä vuorovaikutteinen
  - osallisena **aktori** (toimija, käyttäjä)
    - järjestelmän ulkopuolinen taho, jonka toimintaa järjestelmä ei pysty ohjaamaan
    - voi olla ihmisen rooli, mutta myös toinen järjestelmä
  - voi käyttää hyväkseen toista käyttötapausta, voidaan yhdistellä

© Harri Laine, Jukka Paakki 8

### Käyttötapaukset

- Käyttötapaus on toiminnan abstraktio, joka toteutuu käyttötilanteina. Käyttötilanne voidaan kuvata tilannekuvauksena, **skenaariona**.
- Oven avaus skenaarioina
  1. Liisa näyttää avaintaan lukijalle, järjestelmä tarkistaa pääsyoikeuden ja sytyttää vihreän merkkivalon. Ovi avautuu.
  2. Olli näyttää avaintaan lukijalle, järjestelmä tarkistaa pääsyoikeuden ja huomaa sen vanhentuneen. Punainen merkkivalo syttyy. Olli jää viettämään viikonloppua työpaikalle.
  3. Pekka näyttää avaintaan lukijalle. Lukija ei reagoi mitenkään.
  - Yms.

© Harri Laine, Jukka Paakki 9

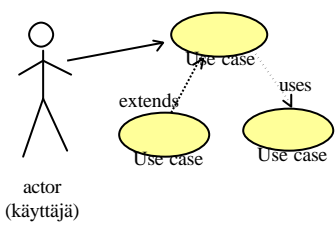
### Käyttötapaukset

- Käyttötapauksesta kuvattava
  - tavoite - **mihin pyritään**
  - aktivointi - **miten käynnistyy**
  - normaalikulku - **se tavallinen tapaus**
  - poikkeustilanteet - **vaihtoehdot, häiriöt, ...**
  - voidaan antaa esimerkkiskenaarioita, mutta skenaariot eivät riitä, vaan niistä on tehtävä yleistys
- Käyttötapaukseen voitava liittää ei-toiminnallisia vaatimuksia - suorituskyky, volyymit

© Harri Laine, Jukka Paakki 10

### Käyttötapaukset

- UML-esittelee käyttötapausmallille graafisen esityksen



actor (käyttäjä)

© Harri Laine, Jukka Paakki 11

### Käyttötapaukset

- Extends-yhteys (laajennus) kytkee poikkeustapaukset ja muunnelmat peruskäyttötapaukseen. Esim.
  - tililäyttö ← tilinlyitys
- Uses-yhteys (käyttö, hyödyntäminen) kuvaa jonkin käyttötapausten käyttöä osana toista käyttötapausta
- Graafisesta esityksestä on hyötyä, jos analyysi on viety niin pitkälle, että mallissa esiintyy runsaasti käyttötapausten välisiä yhteyksiä

© Harri Laine, Jukka Paakki 12

### 2. Tietosisällön kuvaus

- Käyttötapauskuvaus ei mallinna järjestelmän tietosisältöä
- keskeisen tietosisällön kuvaaminen on kuitenkin välttämätöntä useimmissa järjestelmissä
- vaatimusten analysoinnin yhteydessä käytettäviksi soveltuvia tiedonmallinnustekniikoita:
  - luokkakaaviot (UML – Unified Modeling Language)
  - käsitekaaviot (ER – Entity-Relationship)

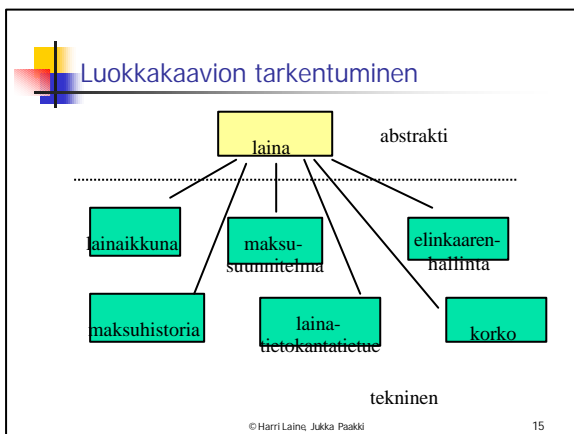
(Oletetaan luokkakaaviot tutuksi asiaksi)

© Harri Laine, Jukka Paakki 13

### Luokkakaaviot

- Luokkakaaviot soveltuvat
  - kuvaamaan tietosisältöä abstraktilla tasolla
    - vaatimusanalyysin kuvaustaso
  - kuvaamaan ohjelmiston konkreettisia tietorakenteita
    - teknisen suunnitelman kuvaustaso
    - teknisiä suunnitelmiakin on eritasoisia, esimerkiksi arkkitehtuurisuunnitelma käsittelee isompia olioita kuin yksityiskohtainen ohjelmasuunnitelma
- Eri tason oliot elelevät omassa ympäristöissään, vaatimustason abstrakti tietosisältöolio voi teknisellä tasolla pirstoutua useaksi olioksi
  - luokkakaaviot tarkentuvat siirryttäessä prosessin vaiheissa eteenpäin

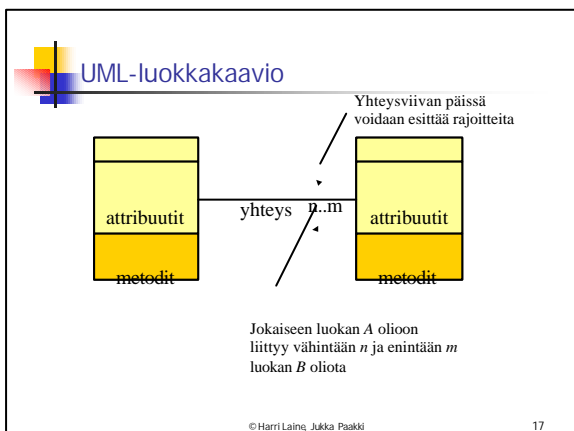
© Harri Laine, Jukka Paakki 14



### Luokkakaavio vaatimusanalyysissä

- Keskeiset tietosisällön osat ja näiden väliset riippuvuudet
- Luokille ei ole yleensä määritelty metodeja, mutta tärkeimmät attribuutit on
- Luokkien väliset yhteydet ovat usein ymmärrettävyyden kannalta oleellisempia kuin attribuutit. Yhteyksiin liittyvät rajoitteet ovat yleensä myös tärkeitä.

© Harri Laine, Jukka Paakki 16



### UML-luokkakaavio

Jokainen luokan  $A$  olio muodostuu joukosta luokan  $B$  olioita. Esim.  $A$  on joukkue ja  $B$  pelaaja. Ei ole esteitä sille, etteikö  $B$  voisi kuulua samanaikaisesti useaan  $A$ -kokonaisuuteen ja siirtyä kokonaisuudesta toiseen.

© Harri Laine, Jukka Paakki 18

### UML-luokkakaavio

Kiinteä kokoonpano: luokan *B* oliot ovat kiinteä osa jotain luokan *A* oliota. *B*-olio on koko elinaikansa osana samaa *A*-oliota. Esim. Näyttökuna - NäyttöKenttä, rakennus - huoneisto. *B*-oliot ovat olemassaoloriippuvia *A*-olioista.

© Harri Laine, Jukka Paakki 19

### UML-luokkakaavio

Luokka *B* perii luokan *A*, *B* on *A*:n aliluokka, *A* on *B*:n ylliluokka : *B*-oliot ovat osajoukko *A*-olioista *B*-oliot ovat erikoistapauksia *A*-olioista, *A*-oliot ovat *B*-olioiden yleistyksiä.

(Toteutustaso: *B*-oliot perivät *A*-luokan koodia.)

© Harri Laine, Jukka Paakki 20

### UML-luokkakaavio

Luokka *B* on 'riippuvuustyyppi' mukaisesti riippuva luokasta *A*. Valmiiksi määriteltäjä riippuvuuksia *uses, instance of, ...*

© Harri Laine, Jukka Paakki 21

### UML-luokkakaavio

- UML-luokkakaaviossa voidaan esittää myös yksittäisiä olioita (luokkanimen tilalla alleviivattu olionimi)
- Luokkakaavioihin voidaan liittää metodien runkoja kommentteina, esimerkiksi pseudokoodina tai ohjelmointikielellä (ei vaatimusanalyysissä)

© Harri Laine, Jukka Paakki 22

### Käyttötapaukset ja luokkakaavio yhdessä

- Käyttötapaukset kuvaavat toimintaa
- Toiminta vaikuttaa tietosisältöön ja/tai riippuu tietosisällöstä
  - tietojen synty, muuttaminen, hyväksikäyttö ja hävittäminen = olioiden/yhteyksien synty, hyväksikäyttö, muuttaminen ja hävittäminen
  - riippuvuuksia: mikä käyttötapaus vaikuttaa luokan olioihin ja miten
  - yhdistelmän kuvaus esimerkiksi riippuvuusmatriisina, katso JSS-moniste

© Harri Laine, Jukka Paakki 23

### Käyttötapaukset ja luokkakaavio yhdessä

- Kuvauksen täydellisyys
  - jokainen käyttötapaus on jotenkin riippuvainen tietosisällöstä
  - käyttötapausmallista löydyttävä käyttötapaukset jokaisen luokan olioiden keskeisten elinkaaritointojen hallintaan

© Harri Laine, Jukka Paakki 24

### 3. Käyttäytymisen kuvaus

- Joissakin järjestelmissä olioiden käyttäytymisen kuvaus on oleellinen vaatimusanalysissa selvitettävä asia
- Olion elinkaari - esimerkiksi puhelun vaiheet, tilauksen käsittelyn työnkulku, sanoman kulku
- Käyttäytymistä on perinteisesti kuvattu **tilakaavioilla** (tila-automaateilla - Laskennan teoria, Ohjelmoinnin ja laskennan perusmallit)

© Harri Laine, Jukka Paakki 25

### Käyttäytyminen

- Käyttäytyminen perustuu siihen, että olio on elinkaarensa aikana erilaisissa **tiloissa** (state)
- Tila** = attribuuttien arvojen ja oliion kytkentöjen muodostama nimetty kokonaisuus, olotila. Tilan määräytymiseen voi vaikuttaa vain yksi attribuutti tai useampien yhdistelmä.
  - Tuoppi on tilassa **täysi**
  - Lasku on tilassa **maksettu**
  - Lompakko on tilassa **tyhjä**
- Perinteisessä tila-automaattimallissa automaatti (tässä olio) voi olla vain **yhdessä tilassa tietyllä hetkellä** ja se vaihtaa tilaansa herätteiden tai täytyvien ehtojen vaikutuksesta

© Harri Laine, Jukka Paakki 26

### Käyttäytyminen

- Siirtymä** (transition) :
  - olion siirtyminen tilasta toiseen
  - siirtymän saa aikaan jokin **heräte** (event, message)
  - heräte on jokin tapahtuma, ja siirtymiseen tilasta toiseen liittyy yleensä oliion suorittamaa toimintaa
  - oliomalleissa heräte on oliion metodikutsu

```

    graph LR
      A[mksamatta] -- maksaminen --> B[maksettu]
    
```

Laskun tiloja, heräte **maksaminen** saa aikaan tilanvaihoksen.

© Harri Laine, Jukka Paakki 27

### Käyttäytyminen

- Siirtymään voidaan liittää ehto, joka säätelee siirtymismahdollisuutta

```

    graph LR
      A[mksamatta] -- maksaminen [eräpäivään alle viikko] --> B[maksettu]
    
```

Laskua ei saa maksaa liian aikaisin.

© Harri Laine, Jukka Paakki 28

### Käyttäytyminen

- Siirtymään voidaan liittää myös toimintaa tai viestin lähetyksen toiselle oliolle

```

    graph TD
      subgraph Lasku
        L1[mksamatta] -- maksaminen [eräpäivään alle viikko] / Lompakko. anna_rahaa(määrä) --> L2[maksettu]
      end
      subgraph Lompakko
        L3[lompakko] -- pullottaa anna_rahaa --> L4[tyhjä]
      end
      L1 -.- L3
    
```

© Harri Laine, Jukka Paakki 29

### Käyttäytyminen

- Alkutila ja mahdolliset lopputilat (laskun elinkaarta)

```

    graph TD
      Start(( )) -- alkutila --> M1[mksamatta]
      M1 -- maksaminen --> M2[maksettu]
      M1 -- eräpäivä oh --> K[karhuttu]
      K -- maksaminen --> M2
      K -- nyt_saa_riittää --> P[perittävänä]
      P -- tuloutus --> M2
      P -- uufet --> L[luottotappiona]
      M2 --> End(( ))
      L --> End
      End -.->|lopputilat| End
    
```

© Harri Laine, Jukka Paakki 30

### Käyttäytyminen

- Tiloihin voidaan liittää toimintaa:
  - tilaan saavuttaessa (entry)
  - tilasta poistuttaessa (exit)
  - tilassa oltaessa (do)

© Harri Laine, Jukka Paakki 31

### Esim: Karkkiautomaatti

© Harri Laine, Jukka Paakki 32

### Käyttäytyminen

- Rakenteinen tila-automaatti (statechart)
  - D. Harel: Statecharts: A Visual Formalism for Complex Systems, Science of Computer Programming, 8, 1987, 231-274
  - Poikkeaa perinteisestä tila-automaatista mm. tarjoamalla
    - rinnakkaiset tilakoneet
      - kokonaisuus voi olla samanaikaisesti useassa rinnakkaisessa tilassa => pienemmät tilakoneet
      - rinnakkaiset tilakoneet voivat olla toisistaan riippumattomia tai riippuvia; esim. olion tilaa voi tarkastella sen eri ominaisuuksien suhteen
    - tilojen historiatiedon

© Harri Laine, Jukka Paakki 33

### Käyttäytyminen

- yleistyshierarkian tiloille - alitila - ylitila

© Harri Laine, Jukka Paakki 34

### Käyttäytyminen

- Edellisen kalvon automaatissa 'help' siirtää mistä tahansa alitilasta ali\_1, ali\_2, ali\_3 tilaan s13
- Automaatti voi olla joko tilassa ali\_1, ali\_2, ali\_3, s13 tai s15
- Jos ulkopuolelta siirrytään ylitilaan, valitaan alitilaksi aloitustilaksi merkitty, eli s15:sta siirrytään ali\_1:een
- Tässä automaatissa on historiatila H, joka tarkoittaa paluuta siihen alitilaan, josta viimeksi poistuttiin, eli tässä palataan help-toiminnon jälkeen samaan tilaan, josta sinne lähdettiin

© Harri Laine, Jukka Paakki 35

### Käyttäytyminen

- Tila-automaateilla on käyttöä muullakin kuin olioiden elinkaarissa, esim.
  - syöttötiedon jäsenitys: saapuva merkki hyväksytään tai hylätään; automaatti voi vaihtaa tilaa, joka taas vaikuttaa seuraavaksi hyväksyttävään merkkiin, ...; mikäli päädytään lopputilaan, on koko merkkijono hyväksytty ja muutoin hylätty
  - käyttöliittymän toiminnan kuvaaminen: tietyt toiminnot mahdollisia vain joissain tiloissa

© Harri Laine, Jukka Paakki 36