

Ohjelmistojen mallintaminen

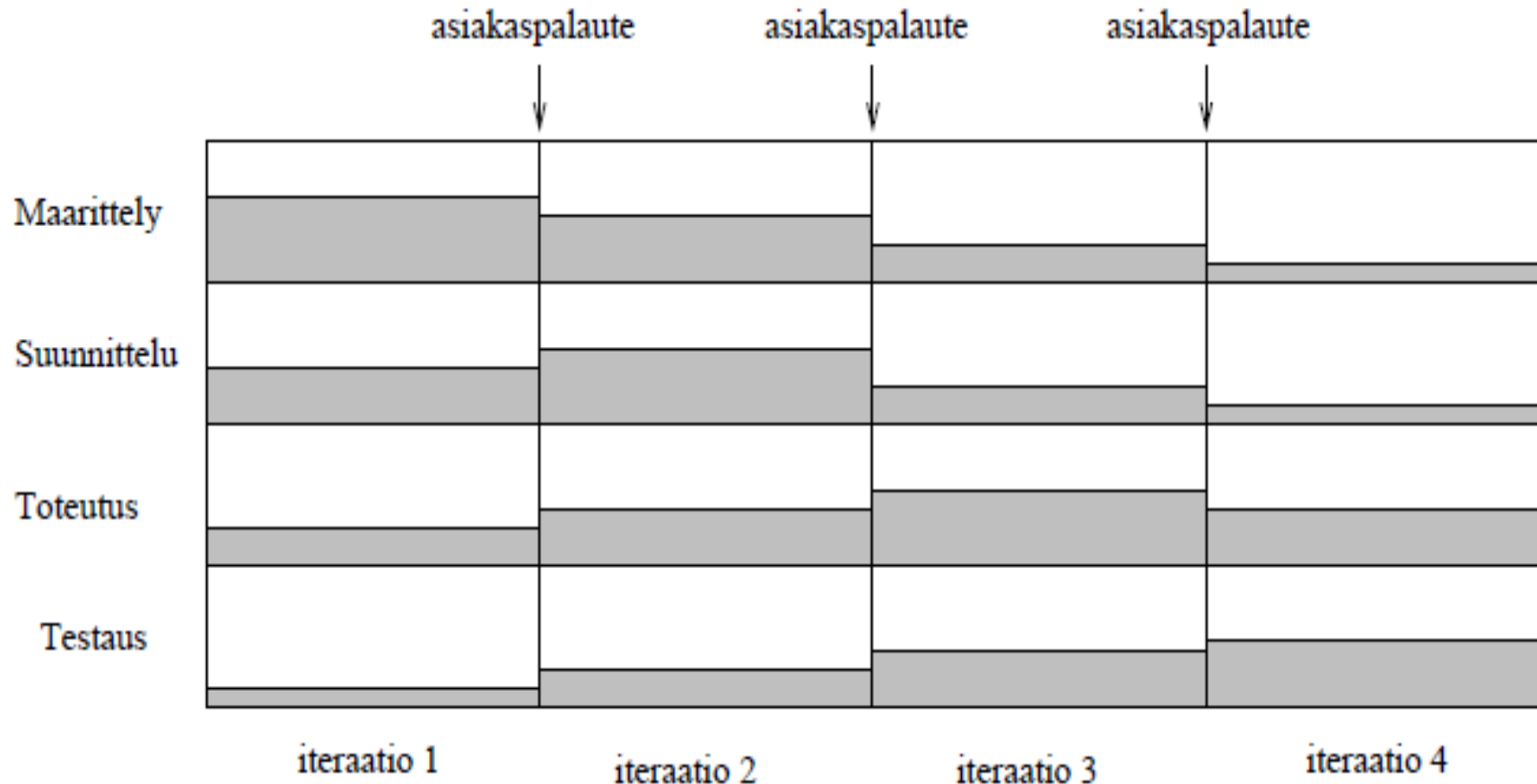
Luento 2, pe 5.11.

Kertausta

- Ohjelmistotuotantoprosessin vaiheet:
 - Vaatimusanalyysi- ja määrittely
 - Mitä halutaan?
 - Suunnittelu
 - Miten tehdään?
 - Toteutus
 - Ohjelmointi
 - Testaus
 - Varmistetaan että toimii niin kuin halutaan
 - Ylläpito
 - Korjataan bugit ja tehdään laajennuksia

Iteratiivinen ja ketterä ohjelmistokehitys

- Toisin kuin vesiputouksessa, vaiheet eivät etene peräkkäin
 - Jokaisessa iteraatiossa mukana kaikkia vaihteita



Nyt tarkastelun alla vaatimusanalyysi ja -määrittely

- Vaatimuksia siis kahdenlaisia
- *Toiminnalliset vaatimukset*
 - Mitä toimintoja ohjelmassa on?
 - Esim. kurssihallintojärjestelmä:
 - Opetushallinto voi syöttää kurssin tiedot järjestelmään
 - Opiskelija voi ilmoittautua valitsemalleen kurssille
 - Opettaja voi syöttää opiskelijan suoritustiedot
 - Opettaja voi tulostaa kurssin tulokset
- *Ei-toiminnalliset vaatimukset (eli ympäristön rajoitteet)*
 - Toteutusympäristö, suorituskykyvaatimukset, ...
- Vaatimusmäärittelyssä ei oteta kantaa ohjelman sisäisiin teknisiin ratkaisuihin, ainoastaan siihen miten toiminta näkyy käyttäjälle
- *Miten toiminnalliset vaatimukset tulisi ilmaista?*

Käyttötapausmalli

- Tänään esiteltävä **käyttötapausmalli** (engl. use case model) on yksi tapa ohjelman *toiminnallisten vaatimusten* ilmaisemiseen
 - Ei-toiminnallisten vaatimusten ilmaisemiseen käyttötapausmalli ei juuri ota kantaa, vaan ne on ilmaistava muuten
 - On olemassa muitakin tapoja toiminnallisten vaatimusten ilmaisuun

- Ohjelmisto tarjoaa käyttäjälleen palveluita
 - Ohjelmiston toimintaa voidaan kuvata määrittelemällä sen tarjoamat palvelut
- Palveluilla on **käyttäjä**
 - Henkilö, toinen järjestelmä, laite yms. taho, joka on järjestelmän ulkopuolella, mutta tekemisissä järjestelmän kanssa
- Käyttäjä voi olla
 - Järjestelmän tiedon hyväksikäyttäjä
 - Järjestelmän tietojen lähde

Käyttäjien tunnistaminen

- Hyvä tapa aloittaa vaatimusmäärittely on tunnistaa/etsiä rakennettavan järjestelmän käyttäjät
- Kysymyksiä jotka auttavat:
 - Kuka/mikä saa tulosteita järjestelmästä?
 - Kuka/mikä toimittaa tietoa järjestelmään?
 - Kuka käyttää järjestelmää?
 - Mihin muihin järjestelmiin kehitettävä järjestelmä on yhteydessä?
- Käyttäjä on oikeastaan **rooli**
 - Missä roolissa toimitaan järjestelmän suhteen
 - Yksi ihminen voi toimia monessa roolissa...

TKTL:n kurssi-ilmoittautumisjärjestelmä

- Käyttäjärooleja
 - Opiskelija
 - Opettaja
 - Opetushallinto
 - Suunnittelija
 - Laitoksen johtoryhmä
 - Tilahallintojärjestelmä
 - Henkilöstöhallintajärjestelmä
- Osa käyttäjistä yhteydessä järjestelmään vain epäsuorasti (esim. Johtoryhmä)
- Osa “käyttäjistä” on muita järjestelmiä
 - Sana käyttäjä ei ole terminä tässä tilanteessa paras mahdollinen

Käyttötapaus, engl. use case

- **Käyttötapaus** kuvaa käyttäjän ohjelman avulla suorittaman tehtävän
 - *miten käyttäjä kommunikoi järjestelmän kanssa* tietyssä käyttötilanteessa
 - Käyttötilanteet liittyvät käyttäjän *tarpeeseen* tehdä järjestelmällä jotain
 - Esim.
 - Kurssi-ilmoittautumisjärjestelmä: *Opiskelijan ilmoittautuminen*
 - Mitä vuorovaikutusta käyttäjän ja järjestelmän välillä tapahtuu kun opiskelija ilmoittautuu kurssille?
- Yksi käyttötapaus on looginen, “isompi” kokonaisuus
 - Käyttötapauksella lähtökohta
 - Ja merkityksen omaava lopputulos (goal)
- Eli pienet asiat, kuten “syötä salasana” eivät ole käyttötapauksia
 - Kyseessä pikemminkin yksittäinen operaatio, joka voi sisältyä käyttötapaukseen

Esim. käyttötapaus opiskelija ilmoittautuu kurssille

- *Käyttäjä:* opiskelija
- *Tavoite:* saada kurssipaikka
- *Laukaisija:* opiskelijan tarve
- *Käyttötapauksen kulku:* Opiskelija tutkii kurssitarjontaa ja valitsee ohjelmiston esittämästä tarjonnasta kurssin ja ryhmän, tunnistautuu ja aktivoi ilmoittautumistoinnin. Opiskelija saa kuittauksen ilmoittautumisen onnistumisesta.
- *Poikkeuksellinen toiminta:* Opiskelija ei voi ilmoittautua täynnä olevaan ryhmään. Opiskelija ei voi ilmoittautua, jos hänelle on kirjattu osallistumiseste.
- *Lisähuomioita:* 4 ruuhkahuippua vuodessa, noin 400 ilmoittautumista ensimmäisen 10 minuutin aikana ilmoittautumisen alkamisesta. Muulloin tapahtumia on vähän

Käyttötapauksen kuvaaminen

- **Kuvataan tekstinä**
- Ei ole olemassa täysin vakiintunutta tapaa kuvaukseen (esim. UML ei ota asiaan kantaa), mutta edellinen ja seuraava sivu sekä muut lähteet näyttävät mallia
- Kuvauksessa mukana usein tietyt osat
 - Käyttötapauksen nimi
 - Käyttäjät
 - Laukaisija
 - Esiehto
 - Jälkiehto
 - Käyttötapauksen kulku
 - Poikkeuksellinen toiminta
- Seuraavalla sivulla käyttötapaus *opiskelija ilmoittautuu kurssille* hieman tarkemmalla tasolla kuvattuna

- *Käyttäjä:* opiskelija
- *Tavoite:* saada kurssipaikka
- *Laukaisija:* opiskelijan tarve
- *Esiehto:* opiskelija on ilmoittautunut kuluvalle lukukaudella läsnäolevaksi
- *Jälkiehto:* opiskelija on lisätty haluamansa ryhmän ilmoittautujien listalle
- *Käyttötapausten kulku:*
 1. Opiskelija aloittaa kurssi-ilmoittautumistoiminnon
 2. Järjestelmä näyttää kurssitarjonnan
 3. Opiskelija tutkii kurssitarjontaa
 4. Opiskelija valitsee ohjelmiston esittämästä tarjonnasta kurssin ja ryhmän
 5. Järjestelmä pyytää opiskelijaa tunnistautumaan
 6. Opiskelija tunnistautuu ja aktivoi ilmoittautumistoiminnon
 7. Järjestelmä ilmoittaa opiskelijalle ilmoittautumisen onnistumisesta.
- *Poikkeuksellinen toiminta:*
 - 4a. Opiskelija ei voi valita ryhmää, joka on täynnä
 - 6a. Opiskelija ei voi ilmoittautua, jos hänelle on kirjattu osallistumiseste.

huomioita

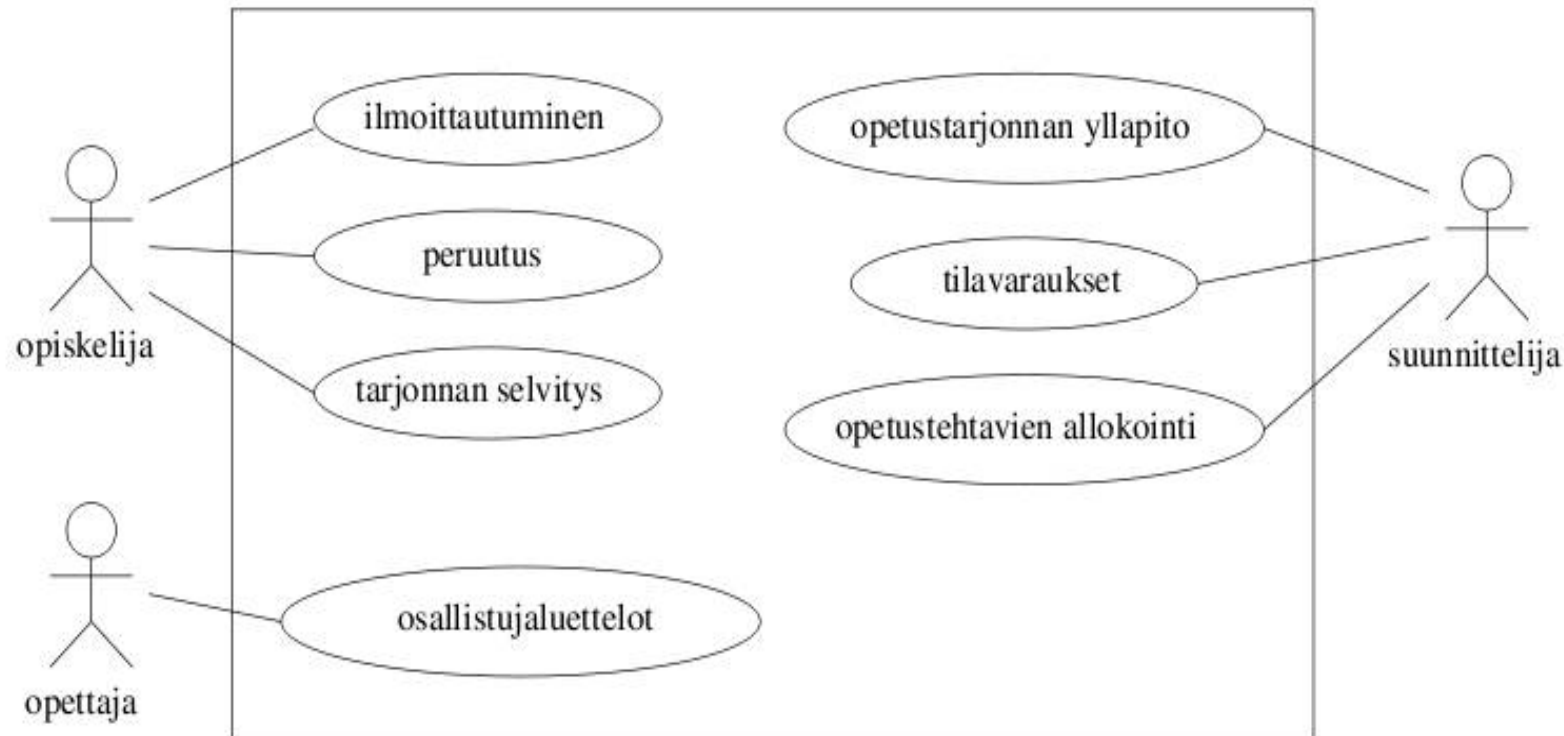
- Esiehto
 - Asioiden tilan joka on vallittava jotta käyttötapaus pystyy käynnistymään
- Jälkiehto
 - Kuvaa mikä on tilanne käyttötapauksen onnistuneen suorituksen jälkeen
- Laukuaisija
 - Mikä aiheuttaa käyttötapauksen käynnistymisen, voi olla myös ajan kuluminen
- Käyttötapauksen kulku
 - Kuvaa onnistuneen suorituksen, usein edellisen sivun tapaan käyttäjän ja koneen välisenä dialogina
- Poikkeuksellinen toimita
 - Mitä tapahtuu jos tapahtumat eivät etene onnistuneen suorituksen kuvauksen mukaan
 - Viittaa onnistuneen suorituksen dialogin numeroihin, esim. jos kohdassa 4 voi tapahtua poikkeus normaaliin kulkuun, kuvataan se askeleena 4a

Käyttötapa *ilmoittautumisen peruminen*

- *Käyttäjä:* opiskelija
- *Tavoite:* perua ilmoittautuminen, välttää sanktiot
- *Laukaisija:* opiskelijan tarve poistaa ilmoittautuminen
- *Esiehto:* opiskelija on ilmoittautunut tietylle kurssille
- *Jälkiehto:* opiskelijan ilmoittautuminen kurssille on poistettu
- *Käyttötapa* *kulku:*
 1. Opiskelija valitsee toiminnon "omat ilmoittautumiset"
 2. Järjestelmä pyytää opiskelijaa tunnistautumaan
 3. Opiskelija tunnistautuu
 4. Järjestelmä näyttää opiskelijan ilmoittautumiset
 5. Opiskelija valitsee tietyn ilmoittautumisensa ja peruu sen
 6. Järjestelmä ilmoittaa opiskelijalle ilmoittautumisen peruuntumisesta

Käyttötapauskaavio

- UML:n **käyttötapauskaavio**n avulla voidaan kuvata käyttötapausten ja käyttäjien (englanniksi termi on actor) keskinäisiä suhteita
- Kurssi-ilmoittautumisjärjestelmän “korkean tason” käyttötapauskaavio



Käyttötapauskaavio

- Käyttäjät kuvataan tikku-ukkoina
 - Olemassa myös vaihtoehtoinen symboli, joka esitellään pian
- Käyttötapaukset järjestelmää kuvaavan nelilön sisällä olevina ellipseinä
 - Ellipsin sisällä käyttötapauksen nimi
- Käyttötapausellipsiin yhdistetään viivalla kaikki sen käyttäjät
 - Kuvaan ei siis piirretä nuolia!
- **HUOM: Käyttötapauskaaviossa ei kuvata mitään järjestelmän sisäisestä rakenteesta**
 - Esim. vaikka tiedettäisiin että järjestelmä sisältää tietokannan, ei sitä tule kuvata käyttötapausmallissa

Käyttötapauskaavion käyttö

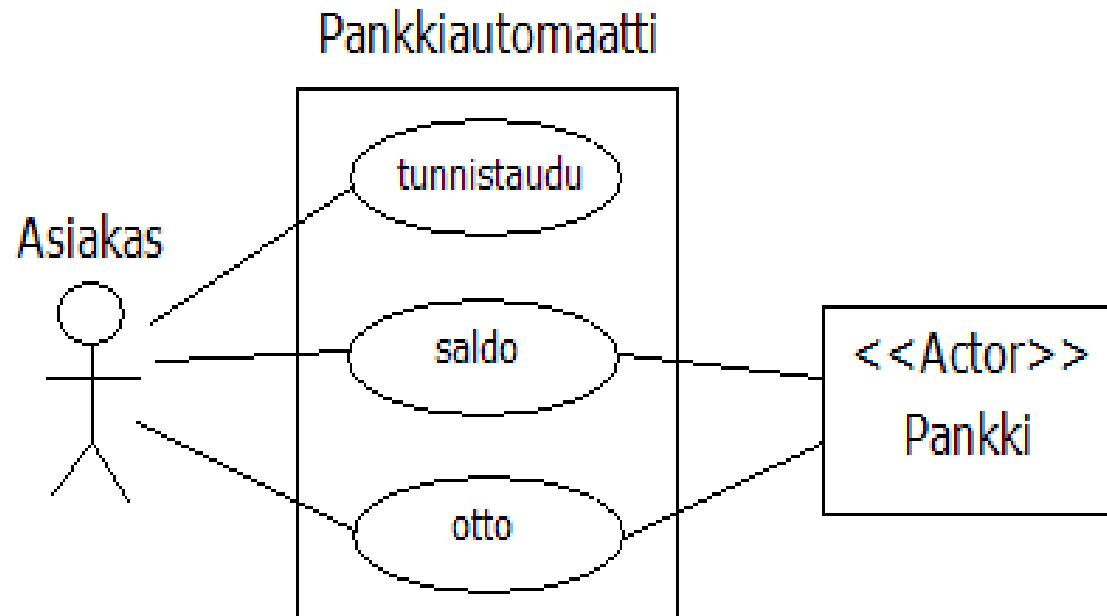
- Kaaviossa siis käyttötapauksista ainoastaan nimi
 - **Käyttötapauksen sisältö kuvataan aina tekstuaalisena esityksenä**
- Kaavio tarjoaa hyvän yleiskuvan järjestelmän käyttäjistä ja palveluista
- **Määrittelydokumentin** alussa kannattaakin olla käyttötapauskaavio “sisällysluettelona”
 - **Jokainen käyttötapaus tulee sitten kirjata tekstuaalisesti tarvittavalla tarkkuudella**

Käyttötapausten dokumentointi

- Ei siis ole olemassa standardoitua tapaa käyttötapauksen kirjaamiseen
- Ohjelmistoprojektissa tulee kuitenkin sopia joku yhteinen muoto, jota kaikkien käyttötapausten dokumentoinnissa noudatetaan
 - *Käyttötapauspohja*
- Edellä esitetyt esimerkkitapaukset tarjoavat mallin käyttötapaukselle
- Internetistä ja kirjoista löytyy myös paljon käyttötapauspohjia
- Ehkä suurimman käyttötapausgurun Alistair Cockburnin käyttötapauspohja löytyy osoitteesta:
 - www.cs.helsinki.fi/u/mluukkai/ohmas10/usecase.pdf

Toinen esimerkki: pankkiautomaatin käyttötapaukset

- Pankkiautomaatin käyttötapaukset ovat *tunnistaudu*, *saldo* ja *otto*
- Käyttötapausten *käyttäjät* eli toimintaan osallistuvat tahot ovat *Asiakas* ja *Pankki*
 - Alla on esitelty tikku-ukolle vaihtoehtoinen tapa merkitä käyttäjä eli laatikko, jossa merkintä <<actor>>
 - Tämä lienee luontevampi jos käyttäjä ei ole ihminen
- Seuraavalla kalvolla käyttötapauksen *Otto* tekstuaalinen kuvaus
 - Huomaa, että esiehto edellyttää, että käyttötapaus tunnistaudu on suoritettu



Käyttötapaus 1: otto

Tavoite: asiakas nostaa tililtään haluamansa määrän rahaa

Käyttäjät: asiakas, pankki

Esiehto: kortti syötetty ja asiakas tunnistautunut

Jälkiehto: käyttäjä saa tililtään haluamansa määrän rahaa

Jos saldo ei riitä, tiliä ei veloiteta

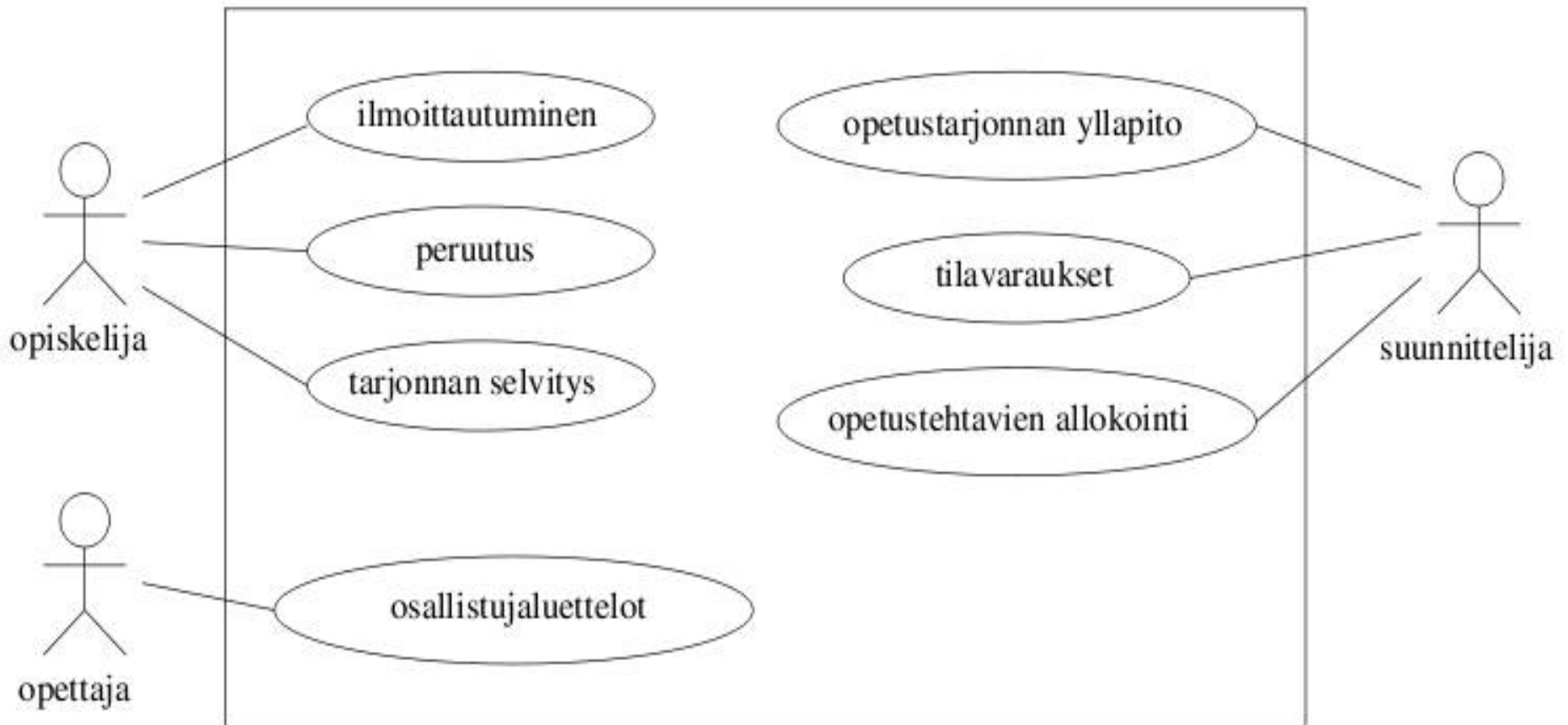
Käyttötapauksen kulku:

- 1 asiakas valitsee otto-toiminnon
- 2 automaatti kysyy nostettavaa summaa
- 3 asiakas syöttää haluamansa summan
- 4 pankilta tarkistetaan riittääkö asiakkaan saldo
- 5 summa veloitetaan asiakkaan tililtä
- 6 kuitti tulostetaan ja annetaan asiakkaalle
- 7 rahat annetaan asiakkaalle
- 8 pankkikortti palautetaan asiakkaalle

Poikkeuksellinen toiminta:

- 4a asiakkaan tilillä ei tarpeeksi rahaa, palautetaan kortti asiakkaalle

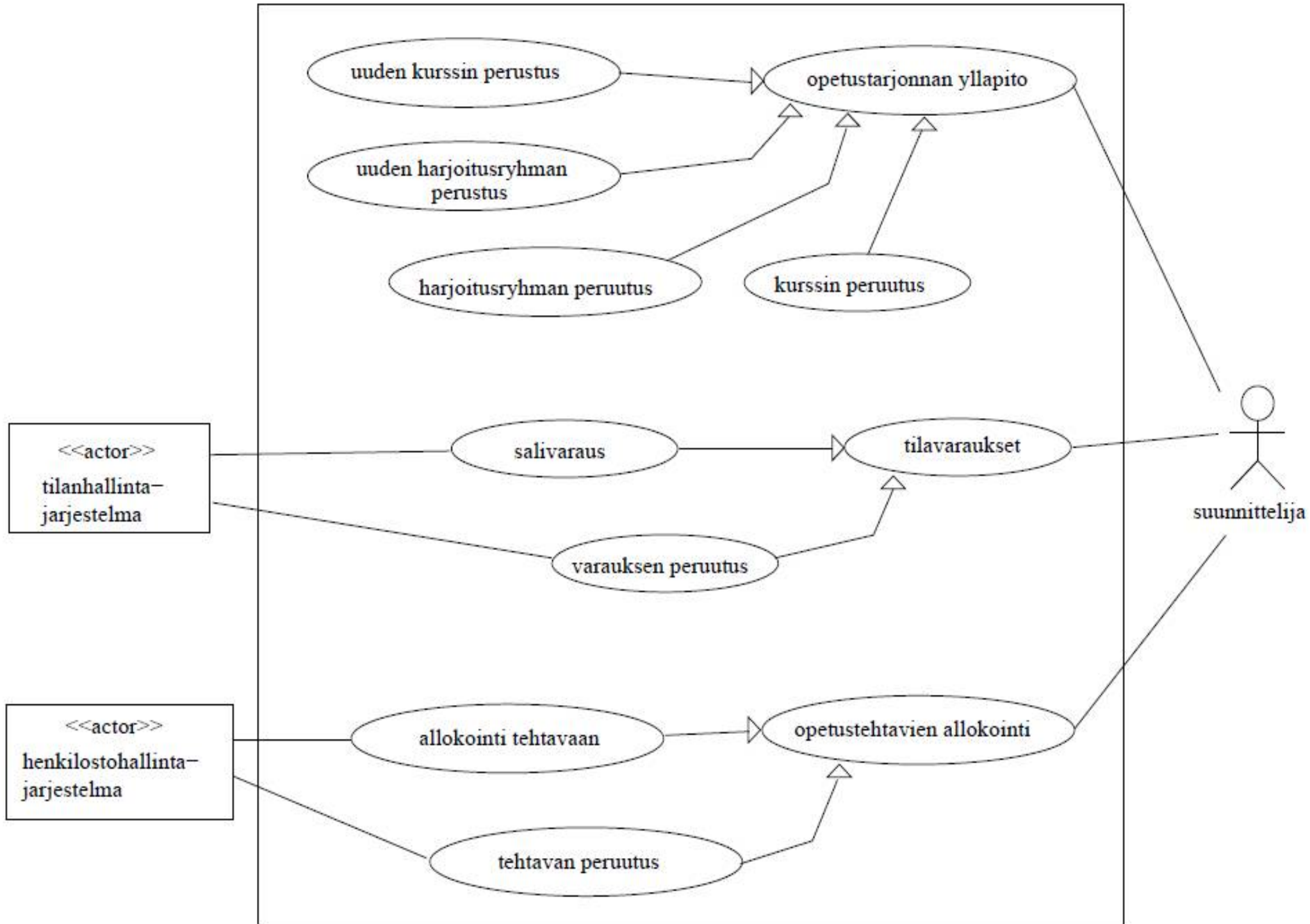
Palataan takaisin kurssi- ilmoittautumisjärjestelmään



Yleistetty ja erikoistettu käyttötapaus

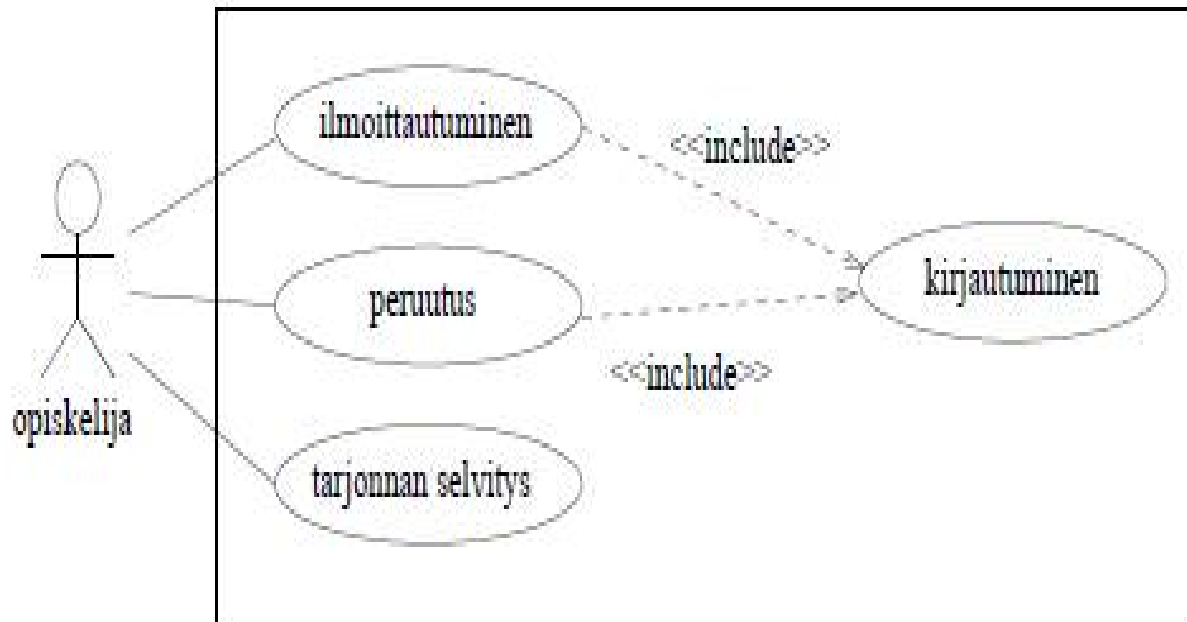
- Suunnittelijan käyttötapauksista erityisesti opetustarjonnan ylläpito on hyvin laaja tehtäväkokonaisuus
- Voidaankin ajatella, että kyseessä on *yleistetty käyttötapaus*, joka oikeasti pitääkin sisällään useita konkreettisia käyttötapauksia, kuten
 - Uuden kurssin perustus
 - Uuden harjoitusryhmän perustus
 - Harjoitusryhmän peruutus
 - Kurssin peruutus
- Seuraavan sivun kaaviossa suunnittelijan käyttötapaukset tarkemmalla tasolla (huomioi miten yleistys merkitään)
- Mukana myös ulkoiset järjestelmät, *tilanhallintajärjestelmä* ja *henkilöstöhallintajärjestelmä*, jotka osallistuvat käyttötapauksiin

Yleistetyt käyttötapaukset käyttötapauskaaviona



Yhteiset osat

- Moneen käyttötapaukseen saattaa liittyä yhteinen osa
- Yhteisestä osasta voidaan tehdä “alikäyttötapaus”, joka *sisällytetään* (include) pääkäyttötapaukseen
- Käyttötapauskaaviossa tätä varten merkintä <<include>>
 - katkoviivanuoli pääkäyttötapauksesta apukäyttötapaukseen
- Esim. käyttötapaus *kirjautuminen* suoritetaan aina kun tehdään *ilmoittautuminen* tai *peruutus*

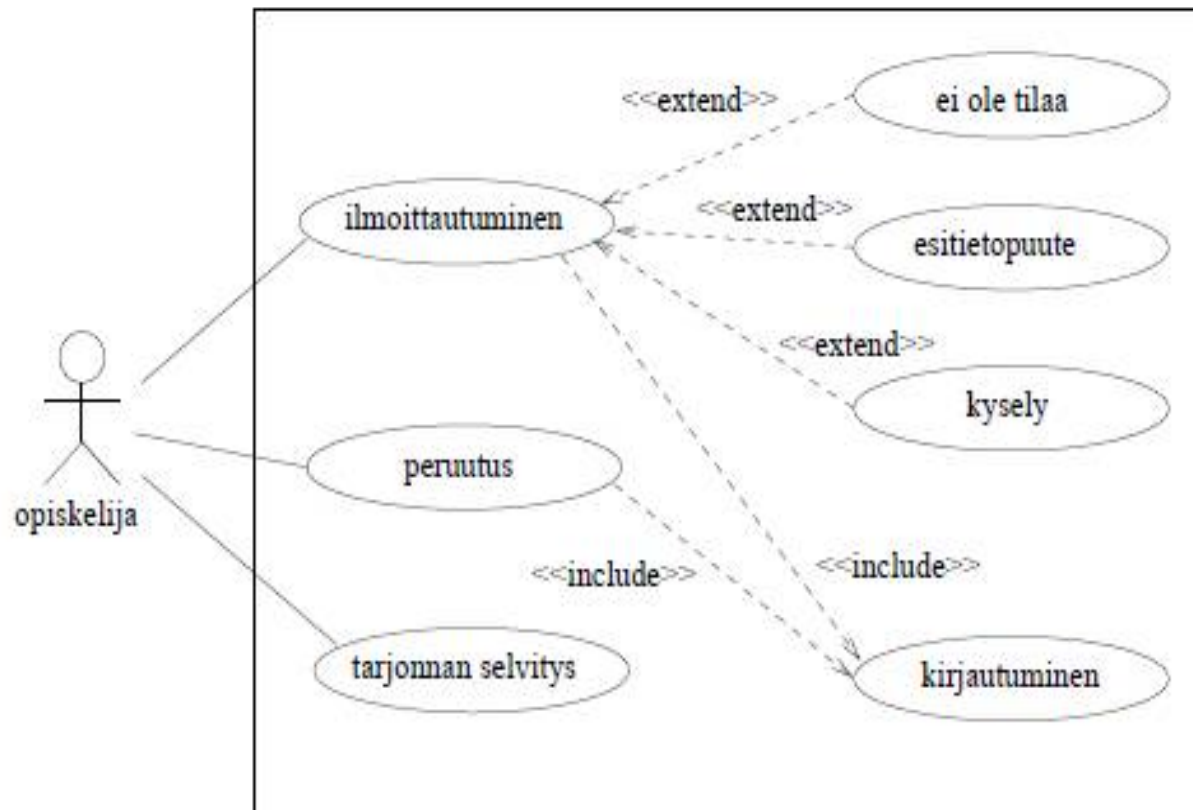


Yhteiset osat ja include

- Apukäyttötapauksen sisällytys on tärkeä ilmaista käyttötapauksen tekstuaalisessa kuvauksessa
 - Muuten ei tietoa missä kohtaan sisällytys tapahtuu
- *Käyttäjä*: opiskelija
- *Tavoite*: saada kurssipaikka
- *Laukaisija*: opiskelijan tarve
- *Esiehto*: opiskelija on ilmoittautunut kuluvalle lukukaudella läsnäolevaksi
- *Jälkiehto*: opiskelija on lisätty haluamansa ryhmän ilmoittautujien listalle
- *Käyttötapauksen kulku*:
 1. Opiskelija aloittaa kurssi-ilmoittautumistoiminnon
 2. Järjestelmä näyttää kurssitarjonnan
 3. Opiskelija tutkii kurssitarjontaa
 4. Opiskelija valitsee ohjelmiston esittämästä tarjonnasta kurssin ja ryhmän
 5. **Suoritetaan käyttötapaus kirjautuminen**
 6. Järjestelmä ilmoittaa opiskelijalle ilmoittautumisen onnistumisesta.
- *Poikkeuksellinen toiminta*:

Poikkeustilanteet ja laajennukset

- Sisällytettävä *käyttötapaus* suoritetaan aina pääkäyttötapauksen suorituksen yhteydessä
- Myös tarvittaessa suoritettava *laajennus tai poikkeustilanne* voidaan kuvata apukäyttötapauksena, joka *laajentaa* (extend) pääkäyttötapausta
 - Laajennus suoritetaan siis vaan *tarvittaessa*
- Esim. Ilmoittautuessa saatetaan huomata esitietopuute, jonka käsittely on oma käyttötapauksensa

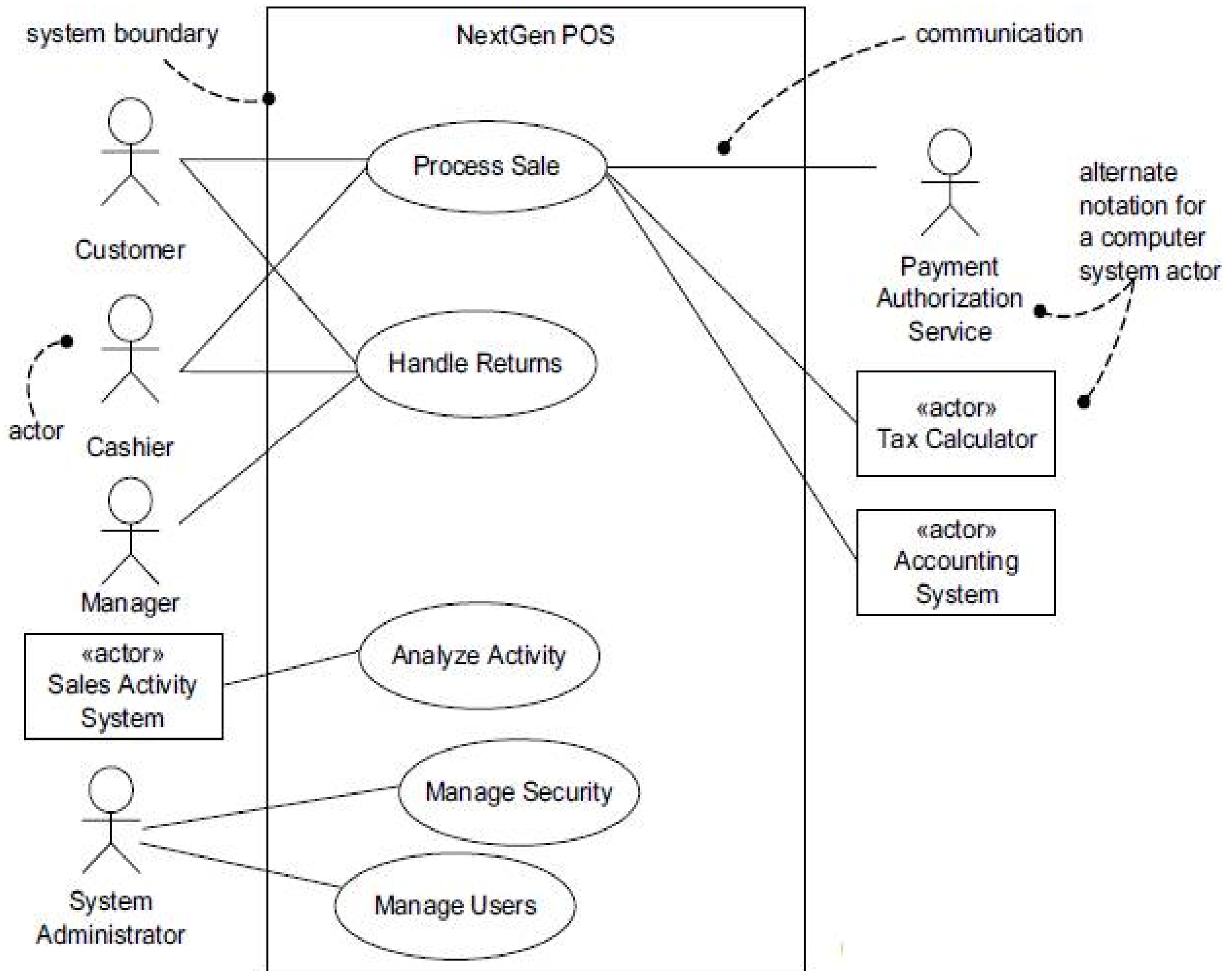


Poikkeustilanteet ja laajennukset

- Huomaa, että laajennuksessa nuolensuunta on apukäyttötapaukseen päin (toisin kuin sisällytyksessä)
- Myös laajennus tulee ehdottomasti merkitä käyttötapauksen tekstuaaliseen kuvaukseen
- Edellisen sivun laajennusesimerkki ei ole erityisen onnistunut
 - Laajennuksienkin pitäisi olla kunnollisia käyttötapauksia (eli asioita joilla on selkeä tavoite), ei metodikutsumaisia kyselyjä tai ilmoituksia (kuten *ei tilaa* - tai *esitietopuute*-ilmoitus)
 - Poikkeustilanteet on parempi kuvata tekstuaalisessa esityksessä ja jättää ne kokonaan pois käyttötapauskaavioista
- Koko laajennuskäsitteen tarve käyttötapauskaavioissa on hieman kyseenalainen

Realistisempi esimerkki: kassapäätejärjestelmä

- Craig Larmanin kirjasta *Applying UML and Patterns*
- Kirjan käyttötapausluku löytyy verkosta
 - <http://www.craiglarman.com/wiki/index.php?title=Articles>
- Aluksi etsitään järjestelmän käyttäjät
- Mietitään käyttäjien tavoitteita: *mitä käyttäjä haluaa saada järjestelmällä tehtyä*
 - Käyttäjän tavoitteellisista toiminnoista (esim. käsittele ostos) tulee tyypillisesti käyttötapauksia
 - Samalla saatetaan löytää uusia käyttäjiä (erityisesti ulkoisia järjestelmiä joihin järjestelmä yhteydessä)
- Hahmotellaan alustava käyttötapausdiagrammi
 - ks. seuraava sivu



Käyttötapauksen tarkentaminen

- Otetaan aluksi tarkasteluun järjestelmän toiminnan kannalta kriittisimmät käyttötapaukset
- Ensin kannattanee tehdä vapaamuotoinen kuvaus käyttötapauksista (“brief use case”)
 - POS, point of sales terminal eli kassapääte

Process Sale: A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

- Tarkempi käyttötapaus kirjoitetaan projektin sopiman käyttötapauspohjan määräämässä muodossa

Use Case UC1: Process Sale

Primary Actor: Cashier

Preconditions: Cashier is identified and authenticated.

Success Guarantee (Postconditions): Sale is saved. Tax is correctly calculated.

Accounting and Inventory are updated. Commissions recorded. Receipt is generated.

Payment authorization approvals are recorded.

Main Success Scenario (or Basic Flow):

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.
Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

- Kuvaus jatkuu: Laajennukset, tarkennukset ja poikkeukset

Extensions (or Alternative Flows):

*a. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

1. System signals error to the Cashier, records the error, and enters a clean state.

2. Cashier starts a new sale.

3a. Invalid identifier:

1. System signals error and rejects entry.

3b. There are multiple of same item category and tracking unique item identity not important (e.g., 5 packages of veggie-burgers):

1. Cashier can enter item category identifier and the quantity.

3-6a: Customer asks Cashier to remove an item from the purchase:

1. Cashier enters item identifier for removal from sale.

2. System displays updated running total.

3-6b. Customer tells Cashier to cancel sale:

1. Cashier cancels sale on System.

3-6c. Cashier suspends the sale:

1. System records sale so that it is available for retrieval on any POS terminal.

- Tarkennuksia:

- 7a. Paying by cash:

1. Cashier enters the cash amount tendered.
2. System presents the balance due, and releases the cash drawer.
3. Cashier deposits cash tendered and returns balance in cash to Customer.
4. System records the cash payment.

- 7b. Paying by credit:

1. Customer enters their credit account information.
2. System sends payment authorization request to an external Payment Authorization Service System, and requests payment approval.
 - 2a. System detects failure to collaborate with external system:
 1. System signals error to Cashier.
 2. Cashier asks Customer for alternate payment.
3. System receives payment approval and signals approval to Cashier.
 - 3a. System receives payment denial:
 1. System signals denial to Cashier.
 2. Cashier asks Customer for alternate payment.
4. System records the credit payment, which includes the payment approval.
5. System presents credit payment signature input mechanism.
6. Cashier asks Customer for a credit payment signature. Customer enters signature.

Tarkkaan kuvattu käyttötapaus

- Esimerkin mallin mukaan käyttötapauksen *pääkulku kannattaa kuvata tiiviisti*
 - Eri askeleiden sisältöä voi tarvittaessa tarkentaa
 - Kuten edellisellä sivulla tarkennettu askel 7 “customer pays...”
- Huomioi tapa, miten poikkeusten ja laajennusten sijainti pääkulussa merkitään
 - 7a => laajentaa/tarkentaa pääkulun kohtaa 7
- Osa jossa laajennukset, tarkennukset ja poikkeukset dokumentoidaan, on usein paljon pidempi kuin normaali kulku
- Koska kyse *vaatimusmäärittelystä*, kuvaus on abstraktilla tasolla
 - *Ei oteta kantaa toteutusyksityiskohtiin*
 - eikä käyttöliittymään
 - Esim. tunnistetaanko ostos viivakoodin perusteella...

Yhteenveto

- Käyttötapaukset ovat yksi tapa kuvata ohjelmiston toiminnallisia vaatimuksia
- Käyttötapauksen tekstuaalinen esitys oleellinen
- Ohjelmistoprojektissa pitää sopia yhteinen tapa (*käyttötapauspohja*) käyttötapausten tekstuaaliseen esitykseen
- Käyttötapauskaavion merkitys lähinnä yleiskuvan antaja
- Jos käytät huomaat käyttäväsi paljon aikaa “oikeaoppisen” käyttötapauskaavion piirtämiseen, ryhdy välittömästi tekemään jotakin hyödyllisempää (esim. käyttötapausten tekstuaalisia esityksiä)