

1 Käyttötapausmalli

Ohjelmisto tarjoaa käyttäjilleen palveluita, jotka perustuvat järjestelmän tietosisältöön. Ohjelmiston toiminta voidaan kuvata määrittelemällä millaisia palveluita ohjelmisto tarjoaa. Käyttötapausmalli (engl. use case model) kuvaa ohjelmiston toimintaa käyttäjän kannalta tarkasteltuna.

Käyttäjä voi olla henkilö, toinen järjestelmä, laite, yms. taho, joka on järjestelmän ulkopuolella, mutta kuitenkin tekemisissä sen kanssa.

Tällainen

taho voi toimia

- * tiedon tuottajana järjestelmään tai
- * tiedon hyödyntäjä

Käyttäjän asemasta käytetään joskus termiä sidosryhmä (engl. stake holder). Käyttäjä voi olla järjestelmän kanssa tekemisissä suoraan tai epäsuorasti.

Käyttäjien tunnistaminen on ensimmäinen tehtävä järjestelmän palveluja määriteltäessä. Käyttäjien löytämiseksi voidaan esittää kysymykset:

- * kuka / mikä saa tulosteita järjestelmästä?
- * kuka / mikä toimittaa tietoa järjestelmään?
- * kuka käyttää järjestelmää?
- * mihin muihin järjestelmiin kehitettävä järjestelmä on yhteydessä?

Näiden kysymysten perusteella tunnistetaan roolit, joissa eri tahot toimivat suhteessa järjestelmään. Nämä roolit määritellään käyttäjiksi.

Esimerkiksi tietojenkäsittelytieteen laitoksen ilmoittautumisjärjestelmän käyttäjärooleja ovat

- * opiskelija
- * opettaja
- * opetushallinto
- * suunnittelija
- * laitoksen johtoryhmä
- * tilahallintojärjestelmä
- * henkilöstöhallintojärjestelmä

Yllä olevista käyttäjärooleista laitoksen johtoryhmä ei ole suorassa yhteydessä järjestelmään. Tilahallintojärjestelmä ja henkilöstöhallintajärjestelmä ovat erillisiä järjestelmiä, joiden palveluita laitoksen opetustietojärjestelmä hyödyntää.

Käyttötapauksella (engl. use case) mallinnetaan käyttäjän ohjelmiston avulla suorittamaa tehtävää eli tapaa käyttää ohjelmistoa. Luonteva tapa henkilökäyttäjien yhteydessä on kytkeä käyttötapaukset käyttäjän työtehtäviin. Tällöin ohjelmiston tarjoama palvelu tukee käyttäjää hänen työtehtävässään. Käyttötapausten laajuus on keskeinen tekijä mallin

ymmärrettävyyden ja hallittavuuden kannalta. Käyttötapaukset eivät saisi määritellä liian laajoja eivätkä myöskään liian suppeita tehtäväkokonaisuuksia. Keskikokoisissa ohjelmistohankkeissa käyttötapauksia on tyypillisesti muutamia kymmeniä ja isommissa hankkeissa jopa satoja.

Pääperiaatteena käyttötapauksia määriteltäessä on, että yhden käyttötapauksen tulisi muodostaa looginen kokonaisuus, jolla on sekä selvä lähtökohta että merkityksen omaava lopputulos. Lähtökohtana eli herätteenä on tapahtuma tai tarve, joka käynnistää käyttötapauksen. Tapahtuma voi olla joko käyttäjän itsensä, jonkin ulkopuolisen tekijän tai järjestelmän aiheuttama. Se voi olla myös ajan kulumisesta aiheutuva.

Kuvassa 1 on hahmoteltu opetukseen ilmoittautumisista vastaavan järjestelmän käyttötapauksia. Opiskelijan käyttötapauksia ovat ilmoittautuminen, peruutus ja tarjonnan selvitys. Tarjonnan selvitys käyttötapauksen lähtökohtana on opiskelijan tarve saada tietoa tarjottavasta opetuksesta luennoista ja harjoituksista. Lopputuloksena opiskelija saa tarvitsemansa tiedot. Ilmoittautuminen käyttötapauksen lähtökohtana on opiskelijan tarve saada opiskelupaikka kurssilta. Lopputuloksena opiskelija kirjataan kurssin osallistujaksi. Peruutus käyttötapauksen lähtökohtana on tarve päästä pois kurssilta. Lopputuloksena on tilanne, jossa opiskelija ei ole enää kirjattu kurssin osallistujaksi.

Figure 1: Ilmoittautumisjärjestelmän käyttötapauksia

On huomionarvoista että tietty taho voi toimia järjestelmän suhteen useammassa erilaisessa käyttäjäroolissa. Esim. laitoksella laskareita pitävä vielä itsekin opiskeleva sivutoiminen opettaja voi toimia ilmoittautumisjärjestelmää käyttäessään välillä opettajan roolissa ja välillä taas opiskelijan roolissa.

Termiä käyttötapaus käytetään yleisesti sekä tyyppitason käsitteenä että ilmentymätason käsitteenä. Tässä yhteydessä noudatetaan UML:n käytäntöä ja käyttötapaus nähdään tehtävätyyppinä. Tietyllä tyyppillä on yleensä useita ilmentymiä eli esimerkkiskenarioita. Käyttötapauksen ilmoittautuminen esimerkkiskenariona voisi olla tapaus Kalle Kenkkunen ilmoittautuu kurssin Ohjelmistojen mallintaminen harjoitusryhmään 2 syksylle 2009.

Käyttötapauksella on käyttäjä (engl. actor), joka käyttötapauksessa toimii vuorovaikutteisesti järjestelmän (ohjelmiston) kanssa toteuttaakseen tavoitteensa. Käyttäjän toiminta on syötteiden antamista ja palautteen saamista. Usein käyttäjä on ihminen, mutta käyttäjä voi olla myös ulkoinen järjestelmä. Käyttötapaukseen liittyy aina tavoite eli asia, jonka käyttäjä haluaa saada aikaan käyttötapauksen avulla. Kuvan 1 opiskelijan käyttötapausten tavoitteena on kussakin tapauksessa saada lähtökohtana mainittu tarve tyydytettyä. Yhdellä käyttötapauksella voi olla myös useita käyttäjiä.

Käyttötapausta kuvattaessa kuvataan käyttäjän ja järjestelmän välinen vuoropuhelu. Kun käyttötapausmallia käytetään määrittelyvaiheessa ja vaatimusanalyysin apuvälineenä kuvauksessa tulisi esittää vain vuoropuhelun sisältö, mitä käyttötapauksessa tapahtuu ja mistä asioista käyttäjä ja järjestelmä vaihtavat tietoa. Käyttöliittymää ei tässä vaiheessa vielä kiinnitetä, joskin joitain periaatteita saattaa olla tarpeen hahmotella,

jotta käyttäjät ymmärtäisivät, mistä on kyse. Miten vuorovaikutus käytännössä tapahtuu, tulisi kiinnittää vasta suunnitteluvaiheessa. Käyttötapauksesta voidaan tällöin laatia yksityiskohtaisempi suunnittelutason kuvaus.

Käyttötapaus kuvataan esittämällä vuoropuhelun peruskulku. Mahdolliset käyttötapausten kulku muuttavat poikkeustilanteet voidaan määritellä erillisinä poikkeavina kulkuina tai käyttötapausta täydentävinä käyttötapauksina. Käyttötapaukseen voi liittyä vaatimuksia koskien suojausta, vasteaikoja, yms.

Käyttötapausten kuvaamiseen ei ole mitään formaalia tekniikkaa, vaan tapaukset kuvataan luonnollisen kielen tekstinä. Vaikka mitään standardoitua tapaa kuvata käyttötapausta ei ole olemassa, jokaisessa ohjelmistoprojektissa kannattaa kuitenkin sopia yhteinen tapa, miten käyttötapausten kuvataan.

1.1 Esimerkkejä

Seuraavassa esimerkkejä ilmoittautumisjärjestelmään liittyen. \

Kurssille Ilmoittautuminen

- * Käyttäjä: opiskelija
- * Tavoite: saada kurssipaikka
- * Laukaisija: opiskelijan tarve
- * Käyttötapausten kulku: Opiskelija tutkii kurssitarjontaa ja valitsee ohjelmiston esittämästä tarjonnasta kurssin ja ryhmän, tunnistauduu ja aktivoi ilmoittautumistoiminnon. 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
- * Esimerkkitapaus: NN ilmoittautuu kurssin XML-metakieli / kevät 2010 harjoitusryhmään 3.

Ilmoittautumisen peruminen

- * Käyttäjä: opiskelija
- * Tavoite: perua ilmoittautuminen, välttää sanktiot
- * Käyttötapausten kulku: Opiskelija tunnistauduu ja valitsee järjestelmän näyttämistä omista ilmoittautumisistaan peruttavan kohteen sekä aktivoi peruutustoiminnon. Järjestelmä ilmoittaa operaation onnistumisesta.
- * Esimerkkitapaus: NN peruu ilmoittautumisensa kurssin XML-metakieli / kevät 2010 harjoitusryhmään 3.

Edellisissä esimerkeissä käyttötapausten kulku on kuvattu melko abstraktilla tasolla, kiinnittämättä käyttäjän ja järjestelmän välistä interaktiota tarkemmin. Näin toimitaan usein järjestelmän määrittelyn alkuvaiheissa, jotta haluttua toiminnallisuutta ei tule kiinnitettyä liian tarkasti

turhain
aikaisessa vaiheessa.

Edetessä järjestelmän vaatimuksien kartoittamisesta kohti suunnitteluvaihetta, halutaan käyttötapauksessa usein tuoda esiin tapahtumien kulku tarkemmalla tasolla. Tällöin tapahtumien kulku on tapana esittää numeroituna dialogina käyttäjän ja järjestelmän välillä. Joskus käyttötapauksille on myös tapana määritellä esiehto (engl. precondition) ja jälkiehto (engl. postcondition). Esiehto kuvaa asioiden tilan, jonka oletetaan olevan voimassa käyttötapauksen alussa. Jälkiehto taas kuvaa asioiden tilan, minkä oletetaan olevan voimassa käyttötapauksen onnistuneen läpikäymisen jälkeen.

Seuraavassa Kurssille ilmoittautuminen -käyttötapaus, jossa tapahtumien kulku ja järjestys on tuotu tarkemmin esiin. Tällä kertaa myös käyttötapauksen esi- ja jälkiehdot on ilmaistu.

Kurssille Ilmoittautuminen, tarkennus

- * 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. Järjestelmä pyytää opiskelijaa tunnistautumaan
 6. Opiskelija tunnistautuu ja aktivoi ilmoittautumistoiminnon
 7. Järjestelmä ilmoittaa opiskelijalle ilmoittautumisen onnistumisesta.
- * Poikkeuksellinen toiminta: Opiskelija ei voi ilmoittautua täynnä olevaan ryhmään. Opiskelija ei voi ilmoittautua, jos hänelle on kirjattu osallistumisesta.

Näin dokumentoituna käyttötapaus kiinnittää jo useita asioita esim. käyttöliittymän suhteen. Kyseessä onkin aiempaa versiota konkreettisempi käyttötapaus.

1.2 Yleistykset, sisällytykset ja laajennokset

Laajoissa järjestelmissä voidaan lähteä liikkeelle käyttäjien työtehtäviin perustuvista käyttötapauksista. Käyttötapauksia analysoitaessa niistä saatetaan löytää yhteisiä osia, jotka voidaan erottaa omiksi käyttötapauksiksi. Myös erilaiset virhe- ja poikkeustilanteet sekä vaihtoehtoiset kulut käyttötapauksessa voidaan erottaa omiksi erillisiksi käyttötapauksiksi. Näin syntyy riippuvuuksia käyttötapausten välille. Käyttötapauksia voi löytyä myös suuria määriä. Tällöin voi yleiskuvan saamiseksi ohjelmiston toiminnosta olla paikallaan koota yhteen käyttötapauksia ja esittää ne yhtenä yleistettynä (eng. generalized) tapauksena.

Kuvassa 1 esitetyistä käyttötapauksista suunnittelijan käyttötapaukset ovat yleistettyjä käyttötapauksia. Esimerkiksi opetustarjonnan ylläpito jakautuu useaan erilliseen tehtävään: uuden kurssin perustaminen, uuden harjoitusryhmän perustaminen, kurssin poistaminen opetustarjonnasta ja harjoitusryhmän peruutus. Ilmoittautumisjärjestelmässä suunnittelijan käyttötapaukset muodostavat oman osajärjestelmänsä ja yksittäiset käyttötapaukset voitaisiin kuvata yksityiskohtaisemmin tämän yhteydessä. Osajärjestelmän käyttötapauskaavio on kuvassa 2. Kuvasta näemme, että erikoistava käyttötapaus, esim. kurssin peruutus on yhdistetty nuolella[1] yleistävään käyttötapaukseen opetustarjonnan ylläpito. Suunnittelijan osajärjestelmän mallissa on otettu mukaan myös muita järjestelmiä tilahallintojärjestelmä ja henkilöstöhallintojärjestelmä, joiden kanssa suunnittelijan käyttämä osajärjestelmä toimii yhteistyössä.

Figure 2: Suunnittelijan osajärjestelmän käyttötapauskaavio

Useat työtehtävät muodostuvat sarjasta peräkkäisiä toimintoja. Käyttäjän tavoite liittyy tällöin usein koko toimintasarjaan eikä yksittäiseen toimintoon vaikka toiminto toisikin tavoitetta jonkin verran lähemmäs. Käyttötapauksena kannattaa tällöin tarkastella koko sarjaa eikä sen askelia.

Kun käyttötapauksia analysoidaan, voidaan löytää usealle tapaukselle yhteisiä osia. Tällöin nämä voi kuvata omina apukäyttötapauksinaan, vaikka ne olisivat toimintasarjan osan kaltaisia. Esimerkiksi opiskelijan ilmoittautumiseen ja ilmoittautumisen perumiseen liittyy molempiin opiskelijan tunnistautuminen. Tätä varten voidaan määritellä käyttötapaukseksi kirjautuminen. Yhdenkään opiskelijan ensisijaisena tavoitteena ei liene kirjautua järjestelmään. Tästä syystä kirjautumista ei voi pitää pääkäyttötapauksena vaan täydentävänä, pääkäyttötapaukseen sisällytettävänä (engl. include) aputapauksena. Kuvassa 3 käyttötapaukset ilmoittautuminen ja peruutus sisällyttävät käyttötapauksen kirjautuminen toiminnallisuuden. Sisällytys on merkitty pääkäyttötapauksesta apukäyttötapaukseen kohdistuvalla katkoviivalla johon on liitetty stereotyyppi eli tarkenne <<include>>. Huomaa taas miten nuolen pää on piirretty. Se on nyt erilainen kun yleistyksen yhteydessä.

Figure 3: Kirjautuminen apukäyttötapauksena ilmoittautumisen ja perumisen osana

Kuten aiemmin todettiin, kuvataan käyttötapauksessa toiminnan peruskulku. Peruskulkunsa mukaisesti esimerkiksi ilmoittautuminen onnistuu aina. Käytännössä näin ei kuitenkaan välttämättä ole. Ilmoittautumiseen liittyy sääntöjä, jotka saattavat estää kurssille pääsyn. Tällaiset poikkeustapaukset voidaan kuvata käyttötapauksen laajennoksina (engl. extend). Kuvassa 4 on ilmoittautumisen laajennoksina esitetty poikkeustapaukset ei-ole-tilaa ja esitietopuute, jotka kumpikin estävät ilmoittautumisen. Laajennoksena kysely kuvataan tilanne, jossa kurssille pääsemiseksi on vastattava kurssikohtaisiin hakukysymyksiin. Laajennos on merkitty laajentavasta käyttötapauksesta pääkäyttötapaukseen kohdistuvalla katkoviivalla johon on liitetty stereotyyppi <<extend>>.

Figure 4: Käyttötapauksen poikkeuksia

Sisällytyksen ja laajennoksen käyttö hämmentää joskus kokeneitakin käyttötapauksen määrittelijöitä. Ero näiden käytössä on siinä, että

käytettäessä sisällytystä apukäyttötapaus liitetään aina pääkäyttötapaukseen. Eli esimerkissämme kirjautuminen suoritetaan aina peruutuksen ja ilmoittautumisen yhteydessä. Laajennos taas liittyy pääkäyttötapaukseen vain tarvittaessa, eli esim. ei-ole-tilaa -käyttötapaus ilmenee ilmoittautumisen yhteydessä ainoastaan joskus. Huomaa kuvista, että nuoli sisällytyksen ja laajennoksen yhteydessä kulkee eri suuntiin. Sisällytyksessä nuoli kulkee apukäyttötapaukseen päin, kun taas laajennoksessa pääkäyttötapaukseen päin.

Useat käyttötapausasiantuntijat (mm. ehkä tunnetuimman käyttötapaussoppaan kirjoittaja Alistair Cockburn [1]) ovat sitä mieltä, että sekavuuksien välttämiseksi käyttötapausten laajennusta ei välttämättä kannata käyttää, vaan parempi tapa ilmaista esim. poikkeukset on dokumentoida ne käyttötapausten tekstuaalisen esityksen yhteydessä.

1.3 Yhteenveto käyttötapauskaavioiden merkinnöistä

UML määrittelee käyttötapausmallille graafisen esitysmuodon. Tätä muotoa on käytetty yllä esitetyissä kuvissa. Käyttötapausmallin symbolit on esitetty kootusti kuvassa 5. Kuvassa on käytetty kommentteja sisältäviä UML-elementtejä. Kommentit ovat käytettävissä kaikissa UML:n kaaviotyypeissä. Poikkeus on valinnainen täydentävä tai muuntava osa jotain laajempaa kokonaisuutta. Itsenäiset tapaukset ovat itsenäisiä peruskäyttötapauksia.

Käyttötapauskaavion informaatioisisältö on melko vähäinen. Lähinnä kaavio tarjoaa yleiskuvan järjestelmän käyttäjistä ja palveluista. Käyttötapauskaaviosta on enemmän hyötyä lähinnä silloin kun on esitettävä käyttötapausten välisiä riippuvuuksia (poikkeuksia ja osia). On kuitenkin tärkeä muistaa, että käyttötapausmallin merkittävin osa on kunkin käyttötapausten sisällön määrittelevä sanallinen kuvaus.

Figure 5: UML:n käyttötapausmallin graafisen esityksen symbolit

Käyttötapausmalli sopii hyvin vuorovaikutteisten ohjelmistojen kuvaamiseen. Tekstikuvausten tarkkuustason valinta on usein hankalaa. Malli ei sovellu kovin hyvin järjestelmien välisen yhteistyön kuvaamiseen. Ulkopuolinen järjestelmä nähdään käyttäjänä ja yhteistyötä pitäisi kuvata käyttäjän kannalta vaikka luonnollisemmalla tuntuisi pitää kehitettävää järjestelmää tällaisessa tilanteessa käyttäjänä.

Käytännössä käyttötapauksia kannattaa ryhmitellä joko käyttämällä yleistettyjä käyttötapauksia tai UML-tekniikkaan sisältyviä pakkauksia, joita käsitellään luvussa None. Ryhmittely voi tapahtua perustuen käyttötapausten käyttäjiin (esim. opettajan ja opiskelijan käyttötapaukset erikseen) tai järjestelmän toiminnallisuuteen (esim. ilmoittautumiseen ja kurssien arvosteluun liittyvät käyttötapaukset erikseen).

Bibliography

[1] Alistair Cockburn. Writing Effective Use Cases. Addison Wesley, 2001.

[1] Huomioi että nuolenpään piirtotapa erikoistamissuhteessa on "avoin kolmio". Kuten kohta huomaamme, on nuolenpäiden piirtotapa määritelty UML:ssä tarkasti.

