Dynaaminen analyysi I

Luento 6

Antti-Pekka Tuovinen

Tavoitteet

- Testitapausten suunnittelun ja suorituksen perusteet
- ”Black-Box” –testitapausten suunnittelu
  - Ekvivalenssiluokat
  - Raja-arvo (reuna-arvo) analyysi

(Dynaaminen) testaaminen

- Ohjelman tai ohjelman osan (test object) suorittamista
  - Tunnetuissa olosuhteissa
  - Tietyillä syötteillä
  - Todeten, vastaako saatu tulos odotettua tulosta

Kokonaisen ohjelman testaus

- Järjestelmä- ja hyväksyntätestauksessa testin kohteena on toimiva (”testausta vaille valmis”) ohjelma
  - Testin suoritus tapahtuu ohjelman käyttöliittymän tai järjestelmärajapinnan kautta
  - Kaikki testien suoritukseen tarvittava toiminnallisuus on jo implementoitu ohjelmaan
- Yleensä tarvitaan kontrolloitu suoritusympäristö ja erikseen tuotettu testidata

Ohjelman osittainen testaus

- Yksikkö- ja integrointitestauksessa testin kohteena on ohjelman *osittainen* toteutus
  - Ei yleensä sellaisenaan suoritettava
  - *Testipeti* tai *–kehys* (testbed) simuloi testattavan komponentin tai alijärjestelmän suoritusympäristöä siten, että haluttujen testitapausten suorittaminen on mahdollista

Testipeti

- Kurssikirja kuva 5-1 *Test bed*, s. 105

Testien suunnittelu

- Tuloksellinen ohjelman tai sen osan testaus vaatii järjestelmällistä ja vaiheittaista työtapaa
  - Kiinnitä testin tavoitteet ja reunaehdot
  - Määrittele yksittäiset testitapaukset
  - Päätä, miten testi suoritetaan
- Muodollisuuden ja dokumentaation määrä riippuu mm. sovellusalueesta, prosessi-käytännöistä, aikarajoitteista jne.

Testin tavoitteet ja reunaehdot

- Mitä pitää testata ja miten?
  - Testausstrategia
  - Vaatimusten täyttymisen osoittaminen
- Esiehdot ja olosuhteet
  - Esim. testidatan saatavuus
- Jäljitettävyys
  - Mitä vaatimuksia testit koskevat?

Testitapausten määrittely

- Syötteiden (test input) määrittely
- Esi- ja jälkiehdot ennen ja jälkeen testitapauksen suorituksen
- Odotetun tuloksen/käyttäytymisen määrääminen

Testien suoritus

- Testitapaukset kootaan testiajoihin ja skenaarioihin, jotka ryhmitellään esimerkiksi testin aiheen tai testin tavoitteiden mukaan
  - Prioriteetit, riippuvuudet, regressiotestitapaukset
- Testaajien allokointi ja testien aikataulutus
- Testiajojen ja skenaarioiden suorituksen automatisointi (”skriptaus”)

Testitapausten suunnittelutekniikat

- Suunnittelutekniikat jaetaan kahteen peruskategoriaan
  - Black box eli ”musta laatikko” -tekniikat
  - White box eli ”lasilaatikko” -tekniikat

Black box

- Testitapausten suunnittelu perustuu testin kohteen spesifikaatioon
  - Toiminnot, käyttäytyminen
- Testin kohteen rakennetta ja implementaatiota ei oteta huomioon
- Testin kohteesta nähdään vain sen ulkoinen käyttäytyminen ja sen tuottamat tulokset annetuilla syötteillä, ei sen sisäistä tilaa (Point of Observation)
- Kohteen käyttäytymistä voidaan kontrolloida vain syötteillä (Point of Control)

White box

- Testin kohteen sisäisen toteutuksen huomioon ottava tekniikka
  - Kontrolli- ja datavuot
- Kohteen sisäistä tilaa havainnoidaan ja sitä voidaan manipuloida testin suorituksen aikana
- Tavoitteena testin kohteen *rakenteen* kattava testaus

Black box vs. White box

- Kurssikirjan kuva 5-2 s. 109
- Black box -tekniikat sopivat kaikkien testaustasojen testien suunnitteluun
- White box -tekniikat sopivat paremmin komponentti- ja integraatiotestien suunnitteluun

Black box -tekniikat

Ekvivalenssiluokat

Raja-arvo analyysi

Ekvivalenssiluokkiin jaon lähtökohdat

- Testin suunnittelun perusteena on testin kohteen *toiminnallinen määrittely*
  - Syötteet, tulosteet, ulkoisesti havaittava käyttäytyminen
- Kurssin alussa nähtiin, että pienenkin funktion täydellinen testaaminen sen kaikilla mahdollisilla syötteillä on mahdotonta
- Testien suunnittelijan on valittava *järkevä joukko syötteitä* testitapauksiinsa
  - Määrä ja (oletettu) kyky löytää vikoja

Ekvivalenssiluokkiin jaon perusidea

- Jokaisen syötteenä olevan erillisen *parametrin* tai syöte-elementin *koko arvoalue* jaetaan *ekvivalenssiluokkiin*
- Jaon perusteena on *oletus*, että testin kohde *käsittelee kaikki samaan ekvivalenssiluokkaan kuuluvat arvot samalla tavoin* - esimerkiksi
  - Tuottaa tulosarvon käyttäen samaa laskentakaavaa
  - Antaa virheilmoituksen väärästä syötteestä

Ekvivalenssiluokkiin jaon perusidea

- Testattaviksi syötearvoiksi riittää siis valita vain *yksi edustaja* kustakin *ekvivalenssiluokasta*
  - Testin suorittaminen jollain toisella *saman ekvivalenssiluokan* arvolla ei aiheuta oleellisesti erilaista testikohteen reaktiota tai käyttäytymistä
- Myös *epävalidit* eli *väärät syötteet* täytyy jakaa ekvivalenssiluokkiin!

Ekvivalenssiluokkiin jako

- Esimerkki kurssikirja, s. 110-111

Ekvivalenssiluokkien määrittely

1. Vaihe – syötteiden ekvivalenssiluokiin jako
   - Määritetään jokaisen syöte-elementin (parametrin, syötekentän) mahdollisten arvojen alue (input domain)
   - Arvoalue jaetaan alustavasti kahteen ekvivalenssiluokkaan
     - Oikeat syötteet, jotka testin kohde käsittelee spesifikaationsa mukaan
     - Väärät syötteet

Ekvivalenssiluokkien määrittely

2. Vaihe – ekvivalenssiluokkien tarkentaminen
   - Tutkitaan testin kohteen spesifikaatiota tarkemmin, ja jokaista eri tavalla käsiteltävää syötearvojoukkoa kohden määritellään uusi ekvivalenssiluokka
   - Kun jako ei enää tarkennu, valitaan kustakin ekvivalenssiluokasta yksi arvo luokan edustajaksi eli testitapauksissa käytettäväksi konkreettiseksi syötearvoksi
     - myös luokan *raja-arvot* on syytä testata!
   - Testitapausten laatu eli kyky löytää virheitä riippuu siitä, miten hyvin eri ekvivalenssiluokat on tunnistettu ja miten luokkien edustajat on valittu

Ekvivalenssiluokkien määrittely

Esimerkki kurssikirjasta s. 112-113

Joukkojen ja diskreettien arvo-alueiden ekvivalenssiluokat

- Ekvivalenssiluokat primitiivisille tietotyypeille löytyvät yleensä melko helposti (esim. luvuille)
- Syötteenä voi olla myös rakenteisia olioita ja dataa tai joukkojen alkioita
- Ekvivalenssiluokat pitää tällöin määritellä tapauskohtaisesti järkevällä tavalla
  - Syöteolioiden niitten ominaisuuksien (property, attribute) perusteella, jotka vaikuttavat olion käsittelyyn testin kohteessa
  - Esim. *matkustajatyypin* vaikutus matkan hinnan laskentaan

Testitapausten määrittely

- Testin kohteella on yleensä useita syötteitä (parametreja), joista jokaisella on vähintään kaksi ekvivalenssiluokkaa (validit ja väärät syötteet) ja jokaisella luokalla yksi tai useampi edustaja
- Konkreettisissa testitapauksissa jokaiselle eri syöteparametrille pitää antaa jokin arvo
- Täytyy siis valita, *mitkä* erillisten syötteiden ekvivalenssiluokkien *edustajat* yhdistetään testitapausten syötteiksi

Syötekombinaatioiden valinnan perussäännöt

- Kaikki *valideja syötteitä* edustavien ekvivalenssiluokkien edustajien kombinaatiot on testattava
  - *Positiiviset* eli *validit testitapaukset*

Syötekombinaatioiden valinnan perussäännöt

- Jokainen *ei-validi* eli *vääriä syötteitä* edustava ekvivalenssiluokka on myös testattava, mutta niiden edustajia *ei saa yhdistää* toisten ei-validien luokkien edustajien kanssa
  - Valitaan siis jokin positiivinen testitapaus, johon vaihdetaan täsmälleen yksi väärä syöte, jolloin saadaan *negatiivinen* eli *ei-validi testitapaus*
- Jos vääriä syötteitä on useita, voi olla vaikea todentaa, minkä syötteen käsittely aiheuttaa havaitun virheilmoituksen tai poikkeuksen
  - Jokin poikkeuskäsittely voi myös jäädä testaamatta!

Kombinaatioiden määrän hallinta

- Jos syöteparametreja on useita ja valideja ekvivalenssiluokkia on monta, kombinaatioiden määrä kasvaa nopeasti
- Määrää voidaan rajoittaa seuraavilla heuristiikoilla
  - Järjestetään ja *priorisoidaan* kombinaatiot k.o. tapausten oletetun esiintymistiheyden mukaan (tyypilliset tapaukset testataan)
  - Suositaan ekvivalenssiluokkien raja-arvoja sisältäviä kombinaatioita
  - Tarkastellaan parametreja pareittain ja valitaan niiden ekvivalenssiluokkien edustajien kaikki *pariyhdistelmät* sisältävät kombinaatiot
  - Jokaisen ekvivalenssiluokan jokainen edustaja on vähintään yhdessä testitapauksessa mukana

Testauksen lopetusehto

- Ekvivalenssiluokkien avulla voidaan määritellä testauksen lopetusehto

EL-kattavuus =

(testattujen EL:ien lukumäärä / EL:ien kokonaismäärä) * 100 %

Ekvivalenssitekniikan arviointia

- Ekvivalenssiluokkien määrittely on kriittinen tekijä testitapausten tehon kannalta
  - Puutteellinen analyysi voi jättää luokkia löytymättä
- Johtaa vaatimuksissa esiintyvien ehtojen ja rajoitusten kattavaan testaukseen
- Minimoi turhia testitapauksia (kustakin ekvivalenssiluokasta poimitaan vain muutama edustaja)
- Voidaan soveltaa kaikilla testaustasoilla
- Ei ota huomioon riippuvuuksia ja vuorovaikutuksia eri syöteparametrien välillä

Raja-arvo analyysi

- Ekvivalenssiluokkia täydentävä tekniikka
- Lähtökohtana havainnot, joiden mukaan vikoja esiintyy usein ekvivalenssiluokkien rajoilta peräisin olevilla syötearvoilla
  - Rajoja ei ole ehkä määritelty selvästi
  - Ohjelmoija on ymmärtänyt väärin, missä luokkien rajat kulkevat
  - Loogisten lausekkeiden koodausvirheet
- Ei voida käyttää parametreille, joiden e-luokilla ei ole reuna-alkioita (joukot)

Raja-arvo analyysi

- Jos e-luokalla on tunnistettava raja (reuna-alkio), testataan
  1) Raja-arvolla
  2) Raja-arvoa lähimpänä olevalla samaan e-luokkaan kuuluvalla arvolla
  3) Raja-arvoa lähimpänä olevalla e-luokan ulkopuolisella arvolla
- Jos varsinaista raja-arvoa ei ole olemassa, testataan 2) ja 3)
- Liukulukuarvoille täytyy määritellä tarkkuus (pienin arvon muutos), jolla raja-arvotestaus tehdään

Raja-arvo analyysi

- Esimerkki kurssikirja s. 122

Vinkkejä raja-arvo analyysiin

- Syötteen pituus on rajoitettu 1 ja 100 alkion välille; testataan 1, 100, 0 ja 101 alkion mittaisilla syötteillä
- Jos testin kohteen tuottamalla tuloksella (output) on tietty arvoalue rajoineen, kannattaa yrittää löytää syötteet, jotka tuottavat raja-arvot sekä niiden lähimmät arvot
- Järjestettyjen kokoelmien ensimmäiset ja viimeiset alkiot ovat kiinnostavia
- Tyhjät kokoelmat ja taulukot ovat hyviä syötteitä
- Samoin täydet kokoelmat ja taulukot

Vinkkejä raja-arvo analyysiin

- Hyvin suuret syötteenä annettavat tietorakenteet ja datamäärät testaavat kohteen käsittelykapasiteetin rajoja
- Hyvin lähellä toisiaan olevat arvot sekä toisistaan hyvin kaukana olevat arvot testaavat numeeristen funktioiden erottelukykyä

Testauksen lopetusehto

RA-kattavuus =

(testattujen RA:jen lukumäärä / RA:jen kokonaismäärä) * 100 %

Raja-arvo analyysin arviointia

- Hyvin tehokas menetelmä yhdessä ekvivalenssiluokkien kanssa käytettynä
  - Vikojen löytymisen todennäköisyys on suurin rajatapausten käsittelyssä
  - Raja-arvot ja niiden lähimmät arvot riittävät teoriassa e-luokkien edustajiksi
- Tekniikka vaikuttaa yksinkertaiselta, mutta syötearvojen määrittely ei ole aina helppoa rajatapausten käsittelyn liipaisemiseksi testikohteessa

Tavoitteet

- Testien suunnittelun ja suorituksen perusteet
- ”Black-Box” –testitapausten suunnittelu
  - Ekvivalenssiluokat
  - Raja-arvo (reuna-arvo) analyysi