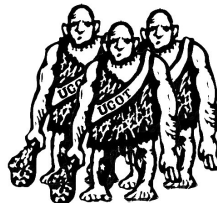


UCOT-Sovellusprojekti

Testausraportti

Ilari Liukko
Tuomo Pieniluoma
Vesa Pikki
Panu Suominen



Versio: 0.06
Julkinen
25. lokakuuta 2006

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2006		
Tilaja	__.__.2006		
Ohjaaja	__.__.2006		

Tietoa dokumentista

Tekijät:

- | | | |
|-------------------------|----------------------|-------------|
| • Ilari Liukko (IL) | ilanliuk@cc.jyu.fi | 050-4367494 |
| • Tuomo Pieniluoma (TP) | tujupien@cc.jyu.fi | 040-7202054 |
| • Vesa Pikki (VP) | vevijopi@cc.jyu.fi | 044-5288031 |
| • Panu Suominen (PS) | panu.suominen@iki.fi | 050-3458484 |

Dokumentin nimi: UCOT-projekti, Testausraportti

Sivumäärä: 5

Tiedosto: UCOT-testausraportti-0.06.tex

Tiivistelmä: Tämän dokumentin tarkoituksena on selvittää UCOT-projektissa käytettävä yleinen testausstrategia ja testiympäristö. Myöhemmin dokumenttiin liitetään iteraatiokohtaisten hyväksyntätestausten ja projektin loppuvaiheen järjestelmätestauksen tulokset.

Avainsanat: Heuristiikka, käyttötapaus, olioluokka, UCOT.

Versiohistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.01	10.10.2006	Testausraportin runko kirjoitettu.	TP
0.02	11.10.2006	Luvun 3 rakennetta muutettu. Virheelliset maininnat testitapauksiin poistettu.	TP
0.03	12.10.2006	Lukua 3 hieman tarkennettu.	TP
0.04	23.10.2006	Lukuja 2, 3.2 ja 4 tarkennettu.	TP
0.05	24.10.2006	Lukuja 2, 3.2 ja 4 tarkennettu hieman lisää. Muutama löytynyt kirjoitusvirhe korjattu.	TP
0.06	25.10.2006	Lisätty riippuvuus java.net pakettiin. Korjattu lukua 4.	TP

Tietoa projektista

UCOT-projekti suunnittelee ja toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle ohjelmiston. Tällä ohjelmistolla voidaan heuristiikkoja käyttäen etsiä käyttötapauksista ohjelmistokehityksen analyysivaiheen olioluokkia.

Tekijät:

- | | | |
|-------------------------|----------------------|-------------|
| • Ilari Liukko (IL) | ilanliuk@cc.jyu.fi | 050-4367494 |
| • Tuomo Pieniluoma (TP) | tujupien@cc.jyu.fi | 040-7202054 |
| • Vesa Pikki (VP) | vevijopi@cc.jyu.fi | 044-5288031 |
| • Panu Suominen (PS) | panu.suominen@iki.fi | 050-3458484 |

Tilaaaja:

- | | | |
|--------------------|--------------------|-------------|
| • Tommi Kärkkäinen | tka@mit.jyu.fi | 040-5677854 |
| • Antti Hakala | anthakal@cc.jyu.fi | 040-7096224 |

Ohjaajat:

- | | | |
|---------------------|--------------------|-------------|
| • Ville Isomöttönen | vilisom@cc.jyu.fi | 014-2604976 |
| • Miika Nurminen | minurmin@cc.jyu.fi | 014-2602530 |

Tarkkailija:

- | | | |
|-------------------|--------------------|-------------|
| • Antti Hallamäki | antahall@cc.jyu.fi | 044-3555356 |
|-------------------|--------------------|-------------|

Yhteystiedot:

- | | |
|---------------------|---|
| • Sähköpostilistat: | ucot06@korppi.jyu.fi,
ucot_opetus@korppi.jyu.fi |
| • Projektiarkisto: | https://korppi.jyu.fi/list-archive/ucot06/ind.html |
| • Opetusarkisto: | https://korppi.jyu.fi/list-archive/ucot_opetus/ind.html |
| • Työhuone: | Ag C222.2 / 014-2604963 |

Sisältö

1	Johdanto	1
1.1	Taustaa	1
2	Testausympäristö	2
3	Testausstrategia	3
3.1	Tarkoitus	3
3.2	Sovellettu yksikkötestaus	3
3.3	Hyväksyntätestaus	3
3.4	Järjestelmätestaus	4
4	Testauksen raportointi	5

1 Johdanto

UCOT-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle ja Agora Centerille ohjelmiston, jolla analyysivaiheen olioluokkia voidaan muodostaa käyttötapauskuvauksista. Sovelluksen tarkoituksena on osittain automatisoida analyysivaiheen olioluokkien suunnittelua. Ohjelmisto toteutetaan Tekes-hankkeen rahoittamalle Tuotanto 2010 -tutkimusprojektille.

Tässä dokumentissa esitellään projektin testaukseen käytettävä strategia, testausympäristö ja testauksen raportoinnissa käytettävät raportointikäytännöt. Iteraatiokohtaisesti suoritettavat hyväksyntätestaukset ja projektin loppuvaiheen järjestelmätestaus raportoidaan tämän dokumentin liitteinä.

1.1 Taustaa

Oliokeskeisen ohjelmistotuotannon kaksi ensimmäistä vaihetta ovat vaatimusmäärittely ja -analyysi. Vaatimusmäärittelyvaiheessa asiakas ja järjestelmää kehittävä osapuoli määrittelevät järjestelmän tarkoituksen ja analysoivat ongelman, jota ratkaisemaan järjestelmä kehitetään. Tuloksena on vaatimusmäärittely ja myös käyttötapauskuvaus, mikäli käytetään käyttötapauspohjaista kehitysmenetelmää. Käyttötapauksista voidaan löytää kirjoitettua tekstiä tutkimalla tärkeimmät sovellusalueen käsitteet, joista tehdään analyysivaiheen olioluokkia. Analyysivaiheen tarkoituksena on tuottaa sovellusaluetta tarkasti kuvaava malli, jonka pohjalta järjestelmä voidaan suunnitella ja toteuttaa.

2 Testausympäristö

UCOT-projektin sovellus kehitetään Javan Standard Editionin versiolla 5.0, joka oli projektin alkaessa uusin projektitilan laitteilla käytössä oleva versio. Tästä johtuen kaikki testaaminen tullaan suorittamaan Java-ympäristön versiossa 5.0. Testauksessa käytettävissä projektitilan laitteissa on käytössä Windows XP Service Pack 2 -käyttäjärjestelmä.

Sovellus ei tule toimimaan vanhemmilla Javan versioilla, koska kehitysvaiheessa käytetään generisiä luokkia. Laitteistoarkkitehtuurikohtaista testausta ei ole mielekästä toteuttaa, koska kyseessä on virtuaalikoneessa ajettava sovellus.

Sovellus on riippuvainen seuraavista Java-kirjastoista:

- java.awt
- java.net
- java.util
- javax.swing
- javax.xml

Näiden lisäksi sovellus käyttää seuraavia ulkoisia komponentteja:

- dot
- Stanford Parser
- PC-Parser

3 Testausstrategia

Tässä luvussa kerrotaan testauksen tarkoituksesta ja käytettävistä menetelmistä.

3.1 Tarkoitus

Testauksen tarkoituksena on varmistaa, että sovellus toteuttaa sille asetetut laadulliset ja toiminnalliset vaatimukset. Sovelluksen tulee säilyttää sen jo toteuttamat toiminnalliset vaatimukset, vaikka lähdekoodia muutettaisiin vielä myöhemmin.

3.2 Sovellettu yksikkötestaus

Hyvin yksityiskohtaista testausta ei projektin laajuuden ja aikataulun vuoksi ehditä toteuttamaan. Tästä johtuen testauksessa keskitytäänkin sovelluksen kannalta tärkeimpiin komponentteihin (*InputAdapter*, *ParserAdapter*, *HeuristicModule*, *Output* ja *Core*). Niiden toimivuus varmistetaan sovelletulla yksikkötestauksella, jossa sovelluksen kriittisimmille komponenteille kirjoitetaan testiyksikkö (JUnit).

Testaaja kirjoittaa sovelluksen suunnitelmien pohjalta komponentille testiyksikön sen toteutuksen aikana. Komponentin testiyksikkö suoritetaan aina, kun kyseisen komponentin lähdekoodia muutetaan tai sovellukseen integroidaan uusia toiminnallisuksia. Jos muutettu komponentti ei läpäise testiä, niin tällöin komponentin muuttaja joko korjaa komponentin sellaiseksi, että se läpäisee testin tai päivittää testin vastaamaan muuttuneita vaatimuksia, jotka eivät enää ole yhteensopivia aiemmin laaditun testiyksikön kanssa. Tällä menetelmällä vältytään turhalta regressiolta tärkeimpien komponenttien kohdalla.

3.3 Hyväksyntätestaus

Iteraatioiden tuotosten toimivuus ja vaatimusten täytyminen varmistetaan iteraatioiden taitekohdissa hyväksyntätestauksella, jossa toteutetut vaatimukset käydään systemaattisesti läpi tilaajan läsnäollessa. Aikataulujen salliessa ohjelmisto asennetaan tilaajan käytettäväksi, jolloin tilaajan on mahdollista antaa projektiryhmälle yksityiskohtaista palautetta ohjelmistosta.

3.4 Järjestelmätestaus

Järjestelmätestauksella tarkoitetaan tämän projektin osalta projektin loppuvaiheeseen sijoittuvaa ad hoc -testausta, käytettävyyden testausta ja lopullista hyväksyntätestausta. Ad hoc -testauksella tarkoitetaan tässä yhteydessä niin sanottua tutkivaa testausta, jossa projektiryhmä testaa sovelluksen eri toimintoja virheiden varalta. Sovelluksen käytettävyydestä osallistuu koko projektiohjausryhmä. Lopullisessa hyväksyntätestauksessa käydään systemaattisesti läpi kaikki sovellukseen toteutetut vaatimukset tilaajan ollessa läsnä.

4 Testauksen raportointi

Iteraatioiden taitekohdissa suoritetuista hyväksyntätesteistä kirjoitetaan erilliset raportit, jotka asetetaan liitteeksi tähän dokumenttiin. Raportista ilmenee iteraatiossa sovellukselle asetetut vaatimukset, niiden hyväksyntä ja mahdollisesti korjattavat virheet. Muussa testauksessa löytyneitä virheitä ei raportoida tässä yhteydessä.

Hyväksyntätestauksen raportin alussa kerrotaan testattu sovelluksen versio ja testauksen päivämäärä. Raportin lopussa kerrotaan iteraation vaatimusten lukumäärä ja niiden täyttymisprosentti. Hyväksyntä kirjataan alla olevan kaltaiseen lomakkeeseen.

Versio		Päivämäärä		
Tunniste	Tärkeys	Käyttötapaus	Vaatus	Hyväksyty
I2.1 (Y4.2)	Pakollinen	K03	Järjestelmä osaa erottaa jäsennetystä vaatimusmäärittelystä entiteetit.	
I2.6	Pakollinen		ParserAdapterin prototyyppi, joka osaa jäsentää luetun käyttötapausten.	
I2.7 (Y5.6)	Pakollinen	K05	Käyttäjä voi muuttaa entiteettien nimiä.	
Vaatuksia yhteensä		Vaatuusten täyttymisprosentti		

Vaatuksut käydään systemaattisesti läpi ja tilaaja merkitsee ne joko hyväksytyiksi tai ehdottaa korjauksia.

Testauksessa löytyneistä virheistä kirjataan ylös siihen liittyvä vaatimus, kuvaus virheestä, mahdollisesti tarvittavat ohjeet virheen toistamiseen, kommentit ja päivämäärä, jolloin virhe korjattiin. Virheet kirjataan erillisiin taulukoihin, jotka liitetään hyväksyntäosion perään.

Luvussa 3.4 kuvatusta lopullisesta hyväksyntätestauksesta raportoidaan vain edellisen hyväksyntätestauksen jälkeen sovellukselle asetetut uudet vaatimukset ja aiemmin korjattavaksi määrätty virheet. Sovelluksen lopullinen hyväksyntä merkitään kokouksen pöytäkirjaan.