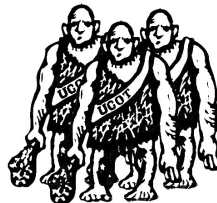


UCOT-Sovellusprojekti

Sovellusraportti

**Ilari Liukko
Tuomo Pieniluoma
Vesa Pikki
Panu Suominen**



Versio: 0.01
Julkinen
7. joulukuuta 2006

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2006		
Tilaaja	__.__.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, Sovellusraportti

Sivumäärä: 12

Tiedosto: sovellusraportti.tex

Tiivistelmä: Tämä dokumentti on sovellusraportti Jyväskylän yliopiston tietotekniikan laitoksen UCOT-sovellusprojektille. Dokumentissa kuvataan UCOT-sovellusprojektin kehittämää ohjelmistoa.

Avainsanat: UCOT, heuristiikka, käyttötapaus, olioluokka.

Versiohistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.01	21.11.2006	Luonnoksen tekeminen aloitettu.	VP
0.02	4.12.2006	Johdanto kirjoitettu ja arkkitehtuurilukua aloitettu.	IL

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
2	Käyttöliittymä	2
3	Arkkitehtuuri ja toiminnallisuus	3
3.1	Yleinen arkkitehtuuri	3
3.1.1	Core	3
3.1.2	UIInterface	4
3.1.3	InputAdapter	4
3.1.4	ParserAdapter	4
3.1.5	HeuristicAdapter	5
3.1.6	OutputAdapter	5
3.2	Abbottin heuristiikka	5
3.3	Stanfordin jäsennin	5
3.4	Dot	5
4	Tiedosto ja tietorakenteet	6
4.1	Käsitelmä	6
4.2	Simple Input Format	6
4.3	ProcessML	6
4.4	GXL	7
5	Rajapinnat	8
5.1	ControllInterface	8
5.2	UIInterface	8
5.3	ParserInterface	8
5.4	HeuristicInterface	8
5.5	OutputInterface	8
5.6	UIInterface	8
6	Ohjelmointikäytännöt ja toteutusympäristö	9
7	Jatkokehitysideat	10
Liitteet		
A	Termit	11

1 Johdanto

UCOT-projekti toteutti Jyväskylän yliopiston tietotekniikan laitokselle, Agora Centerille ja Tekes-hankkeen rahoittamalle Tuotanto 2010-tutkimusprojektille sovelluksen, jolla analyysivaiheen olioluokkia voidaan muodostaa käyttötapauskuvauksista. Sovelluksen tarkoituksena on tukea, sekä osittain automatisoida olioanalyysin tekemistä jäsentämällä käyttötapauskuvausta ja erottelemalla siitä analyysin kannalta oleelliset asiat. Tätä sovelluksen erottamaa, analyysin tukena käytettävää tietoa, kutsutaan käsittemalliksi.

Tämä dokumentti on UCOT-projektin sovellusraportti. Raportissa käydään läpi sovelluksen rakennetta ja osien toimintaa, sekä kuvataan ja perustellaan sovelluksen kehittämisessä käytettyjä toteutusratkaisuja. Lopuksi esitellään projektin aikana esille tulleet kehitysideat, jotka ajanpuutteen vuoksi on jätetty jatkokehitykseen.

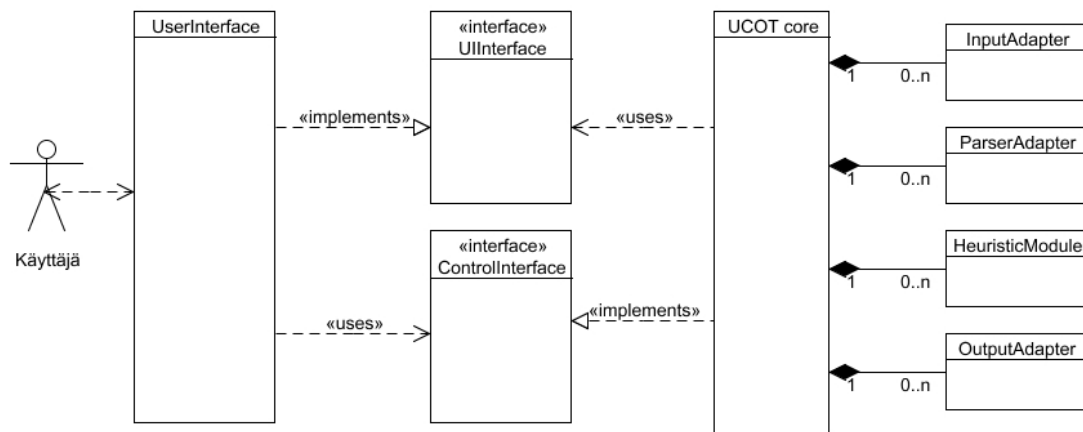
Luvussa 2 esitellään sovelluksen käyttöliittymän rakennetta ja toimintoja. Luvussa 3 esitellään sovelluksen arkkitehtuuria ja toiminnallisuutta, sekä esitellään sovelluksen hyödyntämät ulkoiset komponentit. Luku 4 keskittyy sovelluksen sisäisten, sekä muiden sen tukemien tietorakenteiden esittelyyn. Luvussa 5 esitellään sovelluksen modulin rajapinnat ja luku 6 esittelee sovelluksen lähdekoodin muotoilu ja kommentointi käytänteet, sekä esittelee toteutusympäristön ja projektissa käytetyt työkalut. Luvusta 7 löytyvät jatkokehitysideat sekä toteutusratkaisujen analyysi. Liitteestä A esitellään sovelluksessa ja projektissa esiintyvät termit ja liite B koostuu sovelluksen luokkien Javadoc-kommenteista.

2 Käyttöliittymä

Tässä luvussa esitellään UCOT-sovelluksen käyttöliittymää.

3 Arkkitehtuuri ja toiminnallisuus

Tässä luvussa käydään läpi sovelluksen arkkitehtuuria, sekä sovelluksen osien toimintaa ja tehtäviä. Lisäksi esitellään sovelluksen hyödyntämät ulkopuoliset komponentit ja sovellukset.



Kuva 3.1: Sovelluksen arkkitehtuuri.

3.1 Yleinen arkkitehtuuri

Sovellus suunniteltiin modulaariseksi, jotta sovelluksen komponentteja voidaan myöhemmin korvata tai lisätä tarpeen mukaan.

3.1.1 Core

Sovelluksen toiminnan kannalta keskeisin komponentti on *core*. Sen vastuulla on sovelluksen muiden komponenttien hallinta, sekä niiden toiminnan ohjaaminen. Coren vastuulla on myös muiden komponenttien alustaminen sovelluksen käynnistyessä. Core myös hoitaa sovelluksen *InputAdapterin* lukeman datan ohjaamisen käytettävälle *ParserAdapterille* ja sen ulostulon ohjaamisen *HeuristicAdapterille*.

3.1.2 **UserInterface**

Ohjelman käyttöliittymäkomponentti, jonka kautta käyttäjän on mahdollista ohjata sovelluksen toimintaa.

...ja tarkennusta....

3.1.3 **InputAdapter**

Komponentti, jonka vastuulla on käyttötapauskuvausten lataaminen sille osoitetusta lähteestä. Lähde voi olla kovalevyllä sijaitseva tiedosto, verkossa sijaitseva tiedosto tai jokin vastaava, jonka sijainnin voi osoittaa URL:n avulla.

InputAdapter lataa ja jäsentää coren sille toimittaman lähteen sisältämät käyttötapauskuvaukset sovelluksen sisäiseen esitysmuotoon, jota esitellään tarkemmin luvussa 4. Lataamisen jälkeen InputAdapter toimittaa lähteestä saadut käyttötapauskuvaukset corelle.

3.1.4 **ParserAdapter**

Komponentti, joka suorittaa coren sille sovelluksen sisäisessä esitysmuodossa annetulle jäsentämättömälle käyttötapauskuvaukselle morfologisen jäsennyksen. ParserAdapter palauttaa corelle saamansa käyttötapauskuvauksen jäsennettynä.

ParserAdapter voi itse jäsentää sille annetun käyttötapauskuvauksen tai se voi toimia sovelluksen jäsennin-rajapinnan toteuttavana sovittimena ulkopuoliselle jäsentimelle.

3.1.5 HeuristicAdapter

3.1.6 OutputAdapter

3.2 Abbottin heuristiikka

3.3 Stanfordin jäsennin

3.4 Dot

4 Tiedosto ja tietorakenteet

Esitellään ohjelman sisäisiä tietorakenteita (käsitemalli) ja tiedostorakenteet, joita ohjelma voi lukea ja tallentaa.

4.1 Käsitemalli

— —

Esitellään käsitemallia

- käsitemallin rakenne
- mitä sillä voi kuvata
- kuinka sitä käytetään (käsitemallin editorin rajapinta)

— —

4.2 Simple Input Format

— —

esitellään lyhyesti Simple Input Format

- miksi suunniteltiin
- notaatio
- käyttö

— —

4.3 ProcessML

— —

esitellään ProcessML:ää

- mihin tarkoitukseen suunniteltu
- miten käytetään ohjelmiston kanssa
- esimerkki syntaksista

— —

4.4 GXL

— —

esitellään GXL:ää - mihin tarkoitukseen suunniteltu

- miten käytetään ohjelmiston kanssa
- muita sitä käyttäviä ohjelmistoja
- esimerkki syntaksista

— —

5 Rajapinnat

Esittely rajapinnasta ja niistä toimista mitä rajapinnan toteuttavan luokan tulee myös suorittaa..

5.1 **ControlInterface**

Ytimen rajapinta, jonka kautta UI kutsuu sitä.

5.2 **InputInterface**

Rajapinta käyttötapauksien lataamiseen lähteestä.

5.3 **ParserInterface**

Rajapinta jäsentimien käyttöön.

5.4 **HeuristicInterface**

Rajapinta heuristiikkojen käyttöön.

5.5 **OutputInterface**

Rajapinta käsitelmallin tallentamiseen ulkoiseen muotoon.

5.6 **UIInterface**

Rajapinta käyttöliittymälle, jonka kautta core kutsuu sitä.

6 Ohjelmointikäytännöt ja toteutusympäristö

Kerrotaan lyhyesti kuinka projektin lähdekoodi on muotoiltu ja kommentoitu.
Kerrotaan toteutusympäristöstä.

7 Jatkokehitysideat

Tähän mennessä tulleet ja tulevat ohjelmiston kehittämisideat, joita tämän projektin puitteissa ei pystytä toteuttamaan.

Listataan asioita jotka jälkeinpäin ajateltuna olisi kannattanut tehdä toisin.

A Termit

B Javadoc