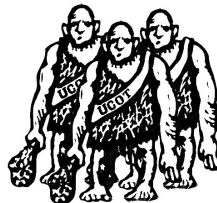


# UCOT-Sovellusprojekti

## Vaatimusmäärittely

Ilari Liukko  
Tuomo Pieniluoma  
Vesa Pikki  
Panu Suominen



Versio: 0.08  
Julkinen  
31. 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, Vaatusmääritys

**Sivumäärä:** 20

**Tiedosto:** UCOT-vaatusmaarittely-0.08.tex

**Tiivistelmä:** Tämä on UCOT-projektin vaatusmääritys. Dokumentti sisältää sovelluksen käyttötapaukset sekä määrittää asetettavat vaatimukset ja rajoitteet, joiden perusteella sovellus suunnitellaan.

**Avainsanat:** Olioanalyysi, UCOT.

## Versiohistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.01	19.9.2006	Ensimmäinen luonnos valmis.	PS
0.02	20.9.2006	Kirjoitettu alustava johdanto, lisätty termejä, lisätty alustavat ohjeet käyttötapauksen kirjoittamista varten yms.	PS
0.03	22.9.2006	Muokattu ja lisätty käyttötapauksia.	PS
0.04	28.9.2006	Lisätty käyttötapauksia ja viittauksia vaatimusmääritykseen.	PS
0.05	2.10.2006	Muutettu käyttötapausluku selvemmäksi, jaoteltu vaatimukset käyttötapauksien mukaisiin ryhmiin.	PS
0.06	9.10.2006	Siirretty käyttötapausluku sovelluksen toiminta-luvun edelle, annettu käyttötapauksille omat tunnistet, vaihdettu viittaukset käyttötapauksiin käyttämään ko. tunnuksia, jaettu syöteen vaatimukset kahteen osaan (Syöte ja Jäsennin), lisätty käyttöliittymän vaatimukset, kirjoitettu 2. iteraation vaatimukset dokumenttiin.	PS
0.07	10.10.2006	Muutettu käyttötapauksien tunnistet, laitettu termit yhteen lukuun.	PS
0.08	31.10.2006	Lisätty 3. iteraation vaatimukset ja 2. iteraation toteutuma.	PS

## Tietoa projektista

UCOT-projekti suunnittelee ja toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle ohjelmiston, jolla voidaan etsiä käyttötapauksista heuristiikkoja käyttäen 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:  | <a href="https://korppi.jyu.fi/list-archive/ucot06/ind.html">https://korppi.jyu.fi/list-archive/ucot06/ind.html</a>           |
| • Opetusarkisto:    | <a href="https://korppi.jyu.fi/list-archive/ucot_opetus/ind.html">https://korppi.jyu.fi/list-archive/ucot_opetus/ind.html</a> |
| • Työhuone:         | AgC 222.2 / 014 2604963   |



# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Käyttötapaukset</b>	<b>2</b>
<b>3</b>	<b>Sovelluksen toiminta</b>	<b>4</b>
3.1	Käyttötapaoksen lukeminen . . . . .	4
3.2	Käyttötapaoksen prosessointi . . . . .	4
3.3	Käsitemallin muokkaaminen . . . . .	4
3.4	Käsitemallin tallentaminen . . . . .	4
<b>4</b>	<b>Vaatimukset</b>	<b>5</b>
4.1	Yleiset vaatimukset . . . . .	6
4.1.1	Syöte . . . . .	6
4.1.2	Jäsennin . . . . .	7
4.1.3	Käyttöliittymä . . . . .	8
4.1.4	Heuristiikka . . . . .	9
4.1.5	Käsitemallin esittäminen ja muokkaaminen . . . . .	10
4.1.6	Tuloste . . . . .	11
4.2	Toteutettavat moduullit . . . . .	11
4.2.1	Syöte . . . . .	11
4.2.2	Heuristiikka . . . . .	11
4.2.3	Tuloste . . . . .	12
4.2.4	Jäsennin . . . . .	12
4.3	Vaatimukset iteraatioille . . . . .	13
4.3.1	1. Iteraatio . . . . .	13
4.3.2	2. Iteraatio . . . . .	13
4.3.3	3. Iteraatio . . . . .	16
<b>Liitteet</b>		
<b>A</b>	<b>Termit</b>	<b>18</b>





# 1 Johdanto

UCOT-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle ohjelmiston, jolla analyysivaiheen olioluokkia voidaan muodostaa käyttötapauskuvauksista. Ohjelman tarkoituksena on tukea olioanalyysin tekemistä jäsentämällä käyttötapauskuvauksista ja erottelemalla siitä analyysin kannalta oleelliset asiat. Tätä ohjelman erottamaa, analyysin tukena käytettävää tietoa, kutsutaan käsitemalliksi.

Tässä dokumentissa esitetään ohjelman yleinen toiminta, esimerkkejä käyttötapauksista ja ohjelmalle asetettavat vaatimukset. Vaatimukset on pyritty rakentamaan tilaajan pyynnöstä käyttötapauksiin pohjautuen.

Luvussa 2 esitellään ohjelman käyttötapauksia, joihin myöhemmin viitataan vaatimuksissa. Tämän jälkeen luvussa 3 käydään läpi ohjelman toiminnan vaiheita hiukan käyttötapauksia tarkemmin ja teknisemmin. Vaatimukset on koottu lukuun 4.

## 2 Käyttötapaukset

Luvussa esitellään käyttötapausten kirjoittamisessa käytetyt periaatteet sekä itse käyttötapaukset, joiden perusteella ohjelman vaatimukset määritellään. Ohjelmiston kehitystä pyritään tekemään käyttötapausten pohjalta, jolloin ohjelma saa samalla myös testiaineistoa. Koska jäsentimien ymmärrys on aina rajallinen ihmiseen verrattuna, on käyttötapausten kirjoittamisessa noudatettava seuraavia sääntöjä:

1. Käytä yksinkertaista kieltä. Mieluiten sellaisia lauseita, jotka noudattavat subjekti-predikaatti-objekti -lauserakennetta.
2. Käytä samaa termiä samalle ohjelman osalle kaikissa käyttötapausten kohdissa ja kaikissa käyttötapauksissa.

### K01 Yleinen käyttötapaus

Käyttötapaus kuvaa ohjelman yleisen toiminnan.

1. Käyttäjä valitsee käyttötapausten jäsenneittäväksi (K02).
2. Ohjelma prosessoi käyttötapausten (K03).
3. Ohjelma esittää käsittemallin (K04).
4. Käyttäjä muokkaa käsittemallia (K05).
5. Ohjelma tallentaa käsittemallin (K06).

### K02 Valitse käyttötapaus

1. Käyttäjä antaa ohjelmalle käyttötapausta sisältävän lähteen.
2. Ohjelma esittää listan lähteen sisältämistä käyttötapauksista.
3. Käyttäjä valitsee käyttötapausten ohjelman esittämästä listasta.

## **K03 Prosessoi käyttötapaus**

Alkuehdot: Käyttötapaus on valittu (K02). Jäsennin on valittu (K07). Heuristiikka on valittu (K08).

1. Ohjelma suorittaa käyttötapausten morfologisen jäsentämisen.
2. Ohjelma antaa morfologisesti jäsennetyn käyttötapausten heuristiikalle.
3. Heuristiikka palauttaa käsittemallin.

## **Käyttötapaukset listana**

Seuraavassa on lista ohjelman käyttötapauksista ja niiden tunnuksista.

**K01 Yleinen käyttötapaus**

**K02 Valitse käyttötapaus**

**K03 Prosessoi käyttötapaus**

**K04 Esitä käsittemalli**

**K05 Muokkaa käsittemallia**

**K06 Tallenna käsittemalli**

**K07 Valitse jäsennin**

**K08 Valitse heuristiikka**

**K09 Valitse lukija**

**K10 Valitse tallennin**

## 3 Sovelluksen toiminta

UCOT-ohjelmiston tarkoitus on tukea olioanalyysin tekijää käyttötapausten tulkitsemisessa. Ohjelmisto lukee käyttötapausten ja antaa käyttäjän muokata saatua käsitelmää. Lopuksi käyttäjällä on mahdollisuus tallentaa työnsä tulokset. Ohjelman toiminta muistuttaa hyvin paljon linjastoa, jossa edellisen ohjelmanosan tulosta käytetään seuraavan syötteenä.

### 3.1 Käyttötapausten lukeminen

Ohjelma lukee syötteen käyttäjän ilmoittamasta lähteestä. Syötteen lukemisesta huolehtii `InputAdapter`. Käyttötapausten muodolla ei ohjelman kannalta saa olla merkitystä, vaan eri muotoisille käyttötapauksille pitää voida kirjoittaa omat lukijansa.

### 3.2 Käyttötapausten prosessointi

`InputAdapter`in tuottama jäsentämätön käyttötapaus jäsennetään ohjelman sisäiseen muotoon `ParserAdapter`in avulla. Tämän jälkeen ohjelma luo *käsitelmän* jäsennetystä käyttötapauksesta. Luonti tapahtuu erilaisia arviointimenetelmiä käyttäen. Arviointi toteutetaan `HeuristicModule`in sisällä.

### 3.3 Käsitelmän muokkaaminen

Koska ohjelman on käytännössä mahdotonta selvittää täydellisesti kaikkea käyttäjän haluamaa tietoa käyttötapauksesta, on käyttäjällä todennäköisesti tarve muokata ohjelman tuottamaa käsitelmää.

### 3.4 Käsitelmän tallentaminen

Käyttäjän ollessa tyytyväinen käsitelmä tallennetaan tulosterausrajapinnan (`OutputInterface`) kautta haluttuun formattiin.

## 4 Vaatimukset

Luvussa esitellään sovellukselle asetettavat toiminnalliset ja tekniset vaatimukset sekä niiden tärkeysasteet. Lisäksi on kerrottu, mihin käyttötapaukseen vaatimus liittyy. Ensiksi on kerrottu yleiset vaatimukset ohjelmistolle, ja tämän jälkeen jokaisen iteraation vaatimukset on eritelty erikseen.

Vaatimukset luokitellaan seuraaviin tärkeysluokkiin:

<b>Pakollinen</b>	Vaatimus on sovelluksen kannalta kriittinen.
<b>Tärkeä</b>	Vaatimuksen toteutuminen antaa sovellukselle huomattavaa lisäarvoa.
<b>Mahdollinen</b>	Vaatimus ei ole kokonaisuuden kannalta oleellinen.
<b>Ajan salliessa</b>	Vaatimus toteutetaan, jos aikaa riittää. Ei oleellinen toiminnallisuus.

Jokaiselle vaatimukselle on oma tunniste, joka näkyy vaatimuksen tunnistekentässä. Vaatimukset on nimetty seuraavaa käytäntöä noudattaen: yleisien vaatimusten tunniste alkaa Y:llä ja iteraatiokohtaisten vaatimusten I-kirjaimella. Kirjainta seuraa pisteellä erotettuna kaksi lukua. Ensimmäinen luku ilmoittaa yleisen vaatimuksen tapauksessa, mihinkä alakategoriaan vaatimus kuuluu. Iteraatioiden kohdalla ensimmäinen numero tarkoittaa iteraation järjestysnumeroa. Pisteiden jälkeen tuleva luku on vaatimuksen numero.

Vaatimuksen numerolla ei ole vaatimuksen toteutusjärjestyksen kannalta mitään merkitystä vaan siitä määrää vaatimuksen tärkeys. Esimerkiksi Y1 . 1 on ohjelman syötteen käsittelijälle asetettu yleinen vaatimus numero 1. I69 . 666 on 69. iteraatiolle asetettu 666. vaatimus.

Käyttötapauskenttä ilmaisee käyttötapauksen tunnisteiden, johon vaatimus liittyy. Osaa vaatimuksista ei voida suoraan johtaa käyttötapauksista. Tällöin käyttötapauskenttä on tyhjä.

## 4.1 Yleiset vaatimukset

Tämä luku sisältää ohjelmalle asetettuja yleisiä vaatimuksia ja rajoitteita, joista vaatimuksia voidaan poimia iteraatioille. Suurin osa vaatimuksista seuraa jostakin käyttötapauksesta.

### 4.1.1 Syöte

Seuraavassa on ohjelman syötteeseen liittyvät vaatimukset.

Tunniste	Tärkeys	Käyttötapaus	Vaatus
Y1.1	Tärkeä	K09	Syöteen lukija (InputAdapter) voidaan vaihtaa.
Y1.2	Pakollinen	K02	InputAdapter palauttaa listan lähteen (URL) sisältämien käyttötapauksien nimistä.
Y1.3	Pakollinen	K03	InputAdapter palauttaa pyydetyn käyttötapauksen jäsentämättömässä muodossa ( <i>jäsentämätön käyttötapaus</i> ).

#### 4.1.2 Jäsennin

Seuraavassa ovat ohjelman syötteen jäsentämiseen liittyvät vaatimukset

Tunniste	Tärkeys	Käyttötapaus	Vaatus
Y2.1	Tärkeä	K07	Jäsennin (ParserAdapter) voidaan vaihtaa.
Y2.2	Pakollinen	K03: 2. askel	ParserAdapter erottaa sanojen sanaluokat.
Y2.3	Tärkeä	K03: 2. askel	ParserAdapter erottaa subjektin, objektin ja predikaatin.

### 4.1.3 Käyttöliittymä

Luku sisältää ohjelman käyttöliittymään liittyvät sekalaiset vaatimukset.

Tunniste	Tärkeys	Käyttötapaus	Vaatus
Y3.1	Pakollinen	K02: 1. askel	Käyttäjä voi valita syötteenä käytetyn lähteen.
Y3.2	Pakollinen	K09	Käyttäjä voi vaihtaa syötteen lukijan. (InputAdapter).
Y3.3	Pakollinen	K07	Käyttäjä voi vaihtaa jäsentimen (ParserAdapter).
Y3.4	Pakollinen	K08	Käyttäjä voi vaihtaa heuristiikan (HeuristicModule).
Y3.5	Pakollinen	K10	Käyttäjä voi vaihtaa tallentimen (OutputAdapter).
Y3.6	Tärkeä	K06	Käyttäjä voi tallentaa käsitemallin.
Y3.7	Mahdollinen		Käyttäjä voi ladata käsitemallin.



#### 4.1.4 Heuristiikka

Heuristiikka hoitaa jäsenetyn käyttötapauksen tulkitsemisen käsitemalliksi.

Tunniste	Tärkeys	Käyttötapaus	Vaatus
Y4.1	Tärkeä	K08	Heuristiikkaa voidaan vaihtaa.
Y4.2	Pakollinen	K03: 3. askel	Heuristiikka erottaa jäsenetystä käyttötapauksesta entiteetit.
Y4.3	Tärkeä	K03: 3. askel	Heuristiikka erottaa jäsenetystä käyttötapauksesta entiteettien vaikutussuhteet.
Y4.4	Tärkeä	K03: 3. askel	Heuristiikka erottaa jäsenetystä käyttötapauksesta omistussuhteet.
Y4.5	Mahdollinen	K03: 3. askel	Heuristiikka erottaa jäsenetystä käyttötapauksesta suhteiden väliset suhteet.

#### 4.1.5 Käsitemallin esittäminen ja muokkaaminen

Vaatimukset seuraavat yleisestä muokausajatuksesta, eli muokkauksen kohteen tietoja voidaan muuttaa, lisätä tai poistaa.

Tunniste	Tärkeys	Käyttötapaus	Vaatus
Y5.1	Pakollinen	K04	Järjestelmä osaa esittää käyttäjälle löydetyt entiteetit.
Y5.2	Pakollinen	K04	Järjestelmä osaa esittää käyttäjälle löydetyt suhteet.
Y5.3	Mahdollinen	K04	Järjestelmä osaa esittää käyttäjälle löydettyjen suhteiden suhteet.
Y5.4	Pakollinen	K05	Käyttäjä voi lisätä entiteettejä.
Y5.5	Pakollinen	K05	Käyttäjä voi poistaa entiteettejä.
Y5.6	Tärkeä	K05	Käyttäjä voi muokata entiteettien nimiä.
Y5.7	Tärkeä	K05	Käyttäjä voi muokata suhteiden nimiä ja tyyppisiä.
Y5.8	Tärkeä	K05	Käyttäjä voi lisätä suhteita.
Y5.9	Tärkeä	K05	Käyttäjä voi poistaa suhteita.
Y5.10	Mahdollinen	K05	Käyttäjä voi ladata uuden käyttötapausten tiedot käsitemalliin.

#### 4.1.6 Tuloste

Tulosteella tarkoitetaan ohjelman tuottamaa tulosta, joko pysyvänä tallenteena tai tulostuksena ruudulle.

Tunniste	Tärkeys	Käyttötapaus	Vaatus
Y6.1	Tärkeä	K10	Tulosteen toteuttaja voidaan vaihtaa.
Y6.2	Pakollinen	K06	Moduuli tallentaa entiteetit.
Y6.3	Pakollinen	K06	Moduuli tallentaa entiteettien väliset suhteet.
Y6.4	Mahdollinen	K06	Moduuli tallentaa suhteiden väliset suhteet.

## 4.2 Toteutettavat moduullit

Luku esittelee erilaisia mahdollisia konkreettisia toteutuksia eri moduulleille.

### 4.2.1 Syöte

Tunnus	Tyyppi	Tärkeys	Kuvaus
T1.1	Rajoitettu teksti	Tärkeä	Osa lukea syntaksiltaan rajoitettuja tekstimuotoisia.
T1.2	Vapaa teksti	Mahdollinen	Osa lukea vapaita tekstimuotoisia käyttötapauksia.
T1.3	ProcessML	Mahdollinen	Osa lukea ProcessML:n mukaisen käyttötapauksen.

### 4.2.2 Heuristiikka

Tunnus	Tyyppi	Tärkeys	Kuvaus
T2.1	Abbott	Pakollinen	Luo käsitellin Abbottin heuristiikan mukaan.

**4.2.3 Tuloste**

Tunnus	Tyyppi	Tärkeys	Kuvaus
T3.1	GLX	Mahdollinen	Käsitemallin tallennus GLX muodossa.

**4.2.4 Jäsennin**

Tunnus	Tyyppi	Tärkeys	Kuvaus
T4.1	Stanford	Tärkeä	Käsitemalli jäsennetään englannin kielestä Stanfordin jäsentimellä.

### 4.3 Vaatimukset iteraatioille

Tässä luvussa esitellään vaatimukset iteraatioittain ja kuinka vaatimukset saatiin toteutettua iteraation päätyttyä. Ensiksi esitellään lyhyesti iteraation tavoitteet ja lisätään sille asetetut vaatimukset. Näiden jälkeen on kerrottu, kuinka tavoitteet saavutettiin.

#### 4.3.1 1. Iteraatio

Ensimmäisen iteraation (22.9. - 6.10.2006) tarkoituksena on esitellä asiakkaalle ohjelman mahdollista toimintaa ja tutkia toimintaketjun toteutuksen mahdollisia ongelmia.

Tunniste	Tärkeys	Käyttötapaus	Vaatimus
I1.1 (Y4.2)	Pakollinen	K03	Järjestelmä osaa erottaa jäsenneytystä vaatimusmäärittelystä entiteetit.
I1.2 (Y4.3)	Pakollinen	K03	Järjestelmä osaa erottaa jäsenneytystä vaatimusmäärittelystä vaikutussuhteet entiteettien välillä.
I1.3 (Y5.1)	Pakollinen	K04	Järjestelmä osaa tulostaa löydetyt entiteetit.
I1.4 (Y5.2)	Pakollinen	K04	Järjestelmä osaa tulostaa löydetyt suhteet.

#### Toteutuminen

Tunniste	Toteutuminen	Kommentit
I1.1	Tehty	
I1.2	Tehty	
I1.3	Tehty	
I1.4	Tehty	

#### 4.3.2 2. Iteraatio

Toisen iteraation (6.10. - 20.10.2006) tavoitteena on saada syöte dynaamisesti vaikakin tiukasti rajatusta lähteestä ja päästä luomaan käyttöliittymää.

Tunniste	Tärkeys	Käyttötapaus	Vaatus
I2.1 (Y4.2)	Pakollinen	K03	Järjestelmä osaa erottaa jäsenneytystä vaatuumääritystä entiteetit.
I2.2 (Y4.3)	Pakollinen	K03	Järjestelmä osaa erottaa jäsenneytystä vaatuumääritystä vaikutus-suhteet entiteettien välillä.
I2.3 (Y5.1)	Pakollinen	K04	Järjestelmä osaa tulostaa löydetyt entiteetit.
I2.4 (Y5.2)	Pakollinen	K04	Järjestelmä osaa tulostaa löydetyt suhteet.
I2.5	Pakollinen		InputAdapterin prototyyppi, joka osaa lukea käyttötapauksia tiedostosta.
I2.6	Pakollinen		ParserAdapterin prototyyppi, joka osaa jäsentää luetun käyttötapauksen.
I2.7 (Y5.6)	Pakollinen	K05	Käyttäjä voi muuttaa entiteettien nimiä.
I2.8 (Y5.4,Y5.5)	Pakollinen	K05	Käyttäjä voi lisätä/poistaa entiteettijä.
I2.9 (Y5.8,Y5.9)	Pakollinen	K05	Käyttäjä voi lisätä/poistaa entiteettien välisiä suhteita.
I2.10 (Y5.7)	Pakollinen	K05	Käyttäjä voi muuttaa suhteiden nimiä/tyyppejä.

**Toteutuminen**

Tunniste	Toteutuminen	Kommentit
I2.1	Tehty	
I2.2	Tehty	
I2.3	Tehty	
I2.4	Tehty	
I2.5	Tehty	
I2.6	Tehty	
I2.7	Tehty	
I2.8	Tehty	
I2.9	Tehty	
I2.10	Osittain	Tyylejä ei voinut muokata.

### 4.3.3 3. Iteraatio

Kolmannen iteraation (20.10. - 10.10.2006) tavoitteena on saada jäsenettyä luonnollisella kielellä kirjoitettu syöte ja parantaa käyttöliittymää. Seuraavien vaatimusten lisäksi myös edellisen iteraation vaatimukset tulee toteuttaa.

Tunniste	Tärkeys	Käyttötapaus	Vaatus
I3.1 (T4.1)	Pakollinen		Stanfordin jäsentimen integrointi binääritasolla.
I3.2 (T1.3)	Pakollinen		ProcessMLää tukeva InputAdapter.
I3.3	Pakollinen		Entiteettien nimet isolla alkukirjaimella.
I3.4	Pakollinen		Metodien nimet pienellä alkukirjaimella.
I3.5	Pakollinen		Käyttöliittymä näyttää ladatut käyttötapaukset.
I3.6	Pakollinen		Käyttötapauksen voi lisätä käsitelmään.
I3.7	Pakollinen		Käyttötapauksen suoritusaskeleet ovat näkyvillä.
I3.8	Pakollinen		Entiteeteille voidaan asettaa entiteettejä attribuuteiksi.
I3.9	Pakollinen		Koko käsitelmän esitetään käyttäen dot-sovellusta.
I3.10	Pakollinen		Käyttäjä voi muokata suhteiden tyyppiä.



**Toteutuminen**

Tunniste	Toteutuminen	Kommentit
I3.1		
I3.2		
I3.3		
I3.4		
I3.5		
I3.6		
I3.7		
I3.8		
I3.9		
I3.10		

## A Termit

<b>Alkuperäinen käyttötapaus</b>	on lähteen sisältämä käyttötapaus.
<b>Core</b>	on sovelluksen ydin, joka ohjaa ohjelman muiden komponenttien toimintaa.
<b>Entiteetti</b>	on vaatimusmäärittelyssä esiintyvä toimija tai toimenpiteen kohde. Käytännössä mikä tahansa substantiivi voi olla entiteetti.
<b>HeuristicCollection</b>	on luokka, joka säilöö <code>HeuristicModule</code> ja.
<b>HeuristicInterface</b>	on rajapinta, joka määrittää miten <code>HeuristicModule</code> en kanssa kommunikoidaan.
<b>HeuristicModule</b>	on luokka, joka suorittaa heuristiikan sille annetulle jäsenetylle käyttötapaukselle ja palauttaa käsitelmän.
<b>InputAdapter</b>	on luokka, joka lataa URL:llä osoitetun lähteen sisältämät alkuperäiset käyttötapaukset ja palauttaa ne jäsentämättöminä käyttötapauksina.
<b>InputInterface</b>	on rajapinta, joka määrittää miten <code>InputAdapter</code> in kanssa kommunikoidaan.
<b>InputCollection</b>	on luokka, joka säilöö input adaptereita.
<b>Iteraatio</b>	tarkoittaa yleisesti jonkin asian toistamista uudelleen siten, että edellisen suorituskerran tulos on seuraavan kerran syöte. Sovelluskehityksessä iteraatiolla tarkoitetaan projektin suorittamista pienissä paloissa edellisen iteraation tulosten toimiesä seuraavan iteraation toteutuksen pohjana. Tuloksilla tässä tapauksessa tarkoitetaan kaikkea ohjelmiston kehityksen tuottamaa materiaalia eikä vain lähdekoodia. Peräkkäiset iteraatiot eivät välttämättä käytä ollenkaan samaa lähdekoodia vaan koodi voidaan välillä kirjoittaa uudestaan.

<b>Jäsennetty käyttötapaus</b>	on käyttötapaus, jolle on suoritettu morfologinen jäsenitys.
<b>Jäsentämätön käyttötapaus</b>	on käyttötapausten suoritusaskeleet tekstimuodossa. Käyttötapaus on jo otettu sisään järjestelmään, mutta sitä ei ole vielä toimitettu parserille.
<b>Käsitelmä</b>	on heuristiikan muodostama malli jäsenetystä käytöstä.
<b>Käyttötapaus</b>	on kuvaus järjestelmän ja sen käyttäjän välisestä vuorovaikutuksesta tietyn tuloksen aikaansaamiseksi.
<b>Käyttötapausten muoto</b>	kertoo, mitä attribuutteja ja missä järjestyksessä sekä muodossa yksittäisen käyttötapausten kuvaus sisältää. Näitä attribuutteja ovat mm. tiedot pääaktorista ja muista aktoreista, tietoa järjestelmän tilasta ennen ja jälkeen käyttötapausten toiminnan sekä käyttötapausten suoritusaskeleet.
<b>LaTeX 2<math>\epsilon</math></b>	on ladontaohjelmisto, millä tämäkin dokumentti on tehty.
<b>Moduuli</b>	on ohjelman osa, joka piilottaa varsinaisen toiminnan toteutuksen sisäänsä. Hyvin kirjoitetun moduulin sisäistä toteutusta on helppo muuttaa. Yleensä moduuli toteuttaa jonkin rajapinnan vaatiman toiminnallisuuden. Esimerkiksi UCOT-ohjelmistossa eri lähteistä tullutta dataa voidaan lukea kun vain datan lukemista varten on toteutettu moduuli, joka täyttää UCOT-ohjelmiston syöterajapinnan määrittelyyn.
<b>Output</b>	on luokka, joka hoitaa heuristiikan tuottaman käsitelmän esittämisen/tallentamisen.
<b>OutputCollection</b>	on luokka, joka säilöo Outputeja
<b>OutputInterface</b>	on rajapinta, joka määrittää outputin kanssa kommunikoidaan.

---

<b>Parser</b>	tarkoittaa morfologista jäsenintä.
<b>ParserAdapter</b>	on luokka joka toteuttaa <code>ParserInterfacen</code> ja kommunikoi parserin kanssa. Ottaa vastaan jäsentämättömän käyttötapauksen ja palauttaa jäsenetyn käyttötapauksen.
<b>ParserCollection</b>	on luokka, joka säilöö <code>ParserAdaptereita</code> .
<b>ParserInterface</b>	on rajapinta, joka määrittää miten <code>ParserAdapterin</code> kanssa kommunikoidaan.
<b>Projekti</b>	tarkoittaa tämän dokumentin yhteydessä sovellusprojektiä.
<b>Rajapinta</b>	erottaa kaksi toisistaan erillistä ohjelman osaa toisistaan siten, että osat tietävät vain osan toisen toiminnallisuudesta. Tällöin rajapinnan takan olevaa osaa voidaan vaihtaa toisen osan häiriintymättä.
<b>Sovellusprojekti</b>	on tietotekniikan laitoksen opintojakso.
<b>Syöte</b>	on ohjelman vastaanottama data.
<b>Syötemoduuli</b>	on ohjelman osa, joka lukee ohjelmalle tarkoitetun syöteen ja palautta ohjelmalle jäsenetyn käyttötapauksen. Rakentuu <code>InputAdapterista</code> ja <code>ParserAdapterista</code> .
<b>Tuloste</b>	on ohjelman tuottama data.
<b>UI</b>	tarkoittaa käyttöliittymää ( <i>user interface</i> ). Tarkentuu myöhemmissä iteraatioissa.
<b>UIInterface</b>	on käyttöliittymän rajapinta.
<b>UCOT</b>	on tämän sovellusprojektin toteuttava ryhmä.
<b>Vaikutussuhde</b>	on kahden entiteetin välillä vallitseva suhde, jossa toinen käyttää toista.