

Dogma

Testaussuunnitelma

ohtu06-dogma-list@cs.helsinki.fi

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Ohjelmistotuotantoprojekti

01/11/2006

Kurssi

581260 Ohjelmistotuotantoprojekti (9 op, 6 ov)

Projektiryhmä

Bjorkgren Joakim
Kuronen Ville
Ristola Toni
Tani Antti
Vihavainen Arto

Asiakas

Niinivaara Olli

Johtoryhmä

Taina Juha
Moilanen Ilari

Kotisivu

<http://cs.helsinki.fi/group/dogma/>

Versiohistoria

<i>Versio</i>	<i>Päiväys</i>	<i>Muutokset</i>
0.1	01.11.2006	Ensimmäinen versio
0.2	06.11.2006	Lisäyksiä
0.3	13.11.2006	Lisäyksiä kappaleeseen 3.2
1.0	16.11.2006	Käyttötapausesimerkki lisätty Testaussuunnitelma jäädytetty

Sisältö

1.Johdanto.....	1
2.Sanasto.....	1
3.Testausstrategia.....	1
3.1 Yksikkötestaus.....	1
3.2 Integroititestaus.....	1
3.3 Järjestelmätestaus.....	2
3.4 Muu testaus.....	2
4.Testausympäristö.....	2
5.Testausvälineet.....	2
6.Testattavat toiminnot.....	3
6.1 Verkkonäkymä.....	3
6.1.1 Resurssin lisääminen uuteen verkkonäkymään.....	3
6.1.2 Resurssin lisääminen vanhaan verkkonäkymään.....	3
6.2 Resurssitaulukko.....	3
6.2.1 Resurssitaulukon järjestäminen attribuuttien mukaisesti.....	4
6.2.2 Resurssitaulukkoon hakeminen tietokannasta.....	4
6.2.3 Lomakenäkymän avaaminen resurssitaulukosta.....	5
6.3 Resurssin tietojen muokkaus.....	5
6.3.1 Resurssin tietojen muokkaus lomakenäkymässä.....	5
6.3.2 Resurssin lisääminen.....	5
6.4 URI-taulukko.....	6
6.4.1 URI:n lisäys.....	6
6.4.2 URI:n poisto.....	6
6.4.3 URI:n avaaminen.....	6
6.5 Tapahtumien seuranta ja tallennus.....	7
6.6 Resurssileikepöytä.....	7
6.6.1 Kopiointi.....	7
6.6.2 Liittäminen.....	7
6.7 Tietokannan valinta.....	8
7.Testattava käyttötapaus.....	8
8.Testausaikataulu.....	9

1. Johdanto

Dogma on metadatataperustainen dokumenttiselain resurssien hallintaan ja visualisointiin. Ohjelma on tarkoitettu käytettäväksi yksityisen henkilön apuna oman resurssitietokantansa käsittelyssä. Käyttäjä voi lisätä tietokantaan resursseja ja hakea niitä sieltä. Dogma tarjoaa myös välineet resurssien ja niiden välisten yhteyksien visualisointiin graafisena verkkona.

2. Sanasto

GUESS	Graafinen ohjelma verkkojen visualisointiin
Haaraumakattavuus %	Kuinka monta % luokan siirtymistä käydään testissa läpi
Kuollut koodi	Luokan osa, mihin ei päästä millään syötteellä
Validointi	Varmistus siitä, että ohjelmisto täyttää sille asetetut vaatimukset
Verifiointi	Varmistus siitä, että ohjelmisto vastaa määrittämiinsä
V-malli	Verifiointi & validointi

3. Testausstrategia

Pääosin Dogman on täytettävä V-mallin, eli verifiointin ja validoinnin perusvaatimukset. On varmistuttava siitä, että ohjelma toteutetaan oikein. Se ei saa kaatuilla, temppuilla tai muuta vastaavaa. Lisäksi on varmistuttava myös siitä, että ohjelmisto vastaa asiakkaan sille asettamia toteutettavia toiminnallisia ja ei-toiminnallisia vaatimuksia.

V-mallin tekniikkana käytetään ryhmän osalta lähinnä testausta, mutta sen lisäksi pidetään myös loppupuolella tarkastuksia ryhmän jäsenten kesken ryhmäkokousten yhteydessä.

3.1 Yksikkötestaus

Yksikkötestauksessa testataan jokaisen toteutetun luokan ja tietokantojen taulukkojen toiminta. Luokat ja taulukot testaavat yksin niistä vastuussa olevat ryhmän jäsenet.

Yksikkötestauksena tarkoituksena on löytää luokasta puutteita, kuollutta koodia, bugeja ja ennen kaikkia varmistaa sen toimivuus.

Yksikkötestauksen suorittaa jokainen ryhmän jäsen luokan toteutuksen jälkeen. Jokaisen jäsenen on varmistuttava siitä, että itse kirjoitetut luokat ja metodit toimivat vaaditulla tavalla.

Yksikkötestauksessa pyritään 100% haaraumakattavuuteen. Tällä tarkoitetaan sitä, että luokan jokainen metodi, rakenne, ehto-lause ja niiden haaraumien sisällöt käydään läpi.

Haaraumakattavuus saadaan kaavalla:

$$(\text{Testattujen siirtymien lkm} / \text{Luokan siirtymien lkm}) * 100 \%$$

Luokasta testataan metodien lisäksi niiden toteutettavien rajapintojen toimivuus.

3.2 Integroititestaus

Integroititestauksella varmistetaan rajapintojen yhtenäinen ja virheetön toimivuus. Löytyvät ongelmat ovat lähinnä rajapintojen väliseen toimivuuteen liittyviä, sillä luokkien sisäiset

ongelmat on jo löydetty yksikkötestauksen aikana.

Integrintitestauksessa käytetään ns. top-down -strategiaa. Tarkoittaen sitä, että ensin rakennetaan ohjelman runko, joka käynnistyy, mutta ei käytännössä tee mitään. Tämän jälkeen runkoon liitetään yksi kerrallaan jo valmiiksi yksikkötestattuja rajapintoja ja varmistetaan niiden yhtenäinen toimivuus. Rajapintoja liitetään runkoon siten, että ne lopulta muodostavat toimivan ohjelmakokonaisuuden.

Dogman runkona toimii resurssitietokanta. Muut osat ovat käytännössä hyödyttömiä ilman toimivaa tietokantaa. Resurssitietokantaan liitetään järjestyksessä muita osia, jotka toimivat sen avulla. Nämä osat ovat järjestyksessä: päivittäjä (mikäli asiakkaalta saatu valmis ja toimiva versio), käyttöliittymä, loki ja verkkonäkymä.

3.3 Järjestelmätestaus

Järjestelmätestaus suoritetaan ryhmän jäsenien kesken integrintitestauksen jälkeen. Lisäksi asiakkaalle annetaan kokeiltavaksi beta-versio ohjelmistosta. Järjestelmätestauksen tarkoituksena on varmistaa asiakkaan ohjelmalle asettamien vaatimuksien oikeellisuus.

Järjestelmätestauksessa noudatetaan ns. V-mallia, eli verifiointia ja validointia. Ryhmän on varmistuttava siitä, että ohjelmisto täyttää sille asetetut implisiittiset ja eksplisiittiset vaatimukset.

3.4 Muu testaus

Käyttöliittymä pyritään beta-testaamaan asiakkaalla. Dogman käyttöliittymä perustuu pitkälti asiakkaan omaan malliin, joten sen sopiminen asiakkaan käyttöön on oleellista. Jos Dogman käyttöliittymässä ilmenee ongelmia, pyritään sitä muokkaamaan selkeämpään suuntaan.

Asiakas suorittaa myös Dogman lopullisen testauksen. Dogma on yksityisen asiakkaan tilaus, vaikka sitä voikin soveltaa yleiseen resurssien hallintaan. Tästä johtuen on tärkeää, että asiakas katsoo vaatimustensa toteutuneen ennen projektityön palauttamista. Ryhmän jäsenten on seurattava asiakkaan toimintaa lopullisen testauksen aikana ja otettava huomioon mahdolliset ongelmakohdat. Ongelmakohdat korjataan, mikäli ne ovat kriittisiä.

Lopullinen testaus on valmis, kun mahdolliset ongelmakohdat on korjattu ja asiakas on hyväksynyt toimivan version ohjelmasta.

4. Testausympäristö

Vaatimusdokumentin mukaisesti Dogma testataan tietojenkäsittelytieteen laitoksen linux-käyttöjärjestelmän KDE-ympäristössä.

5. Testausvälineet

Yksikkötestaus suoritetaan JUnitin avulla. Eclipse tarjoaa suoran tuen JUnitin käyttöön, joten Eclipseä käytetään myös.

6. Testattavat toiminnot

Dogman vaatimukset perustuvat pääsääntöisesti toiminnallisiin ja ei-toiminnallisiin vaatimuksiin, joten myös testaus perustuu niihin.

6.1 Verkkonäkymä

Selite: käyttäjä pystyy avaamaan haluamansa tulosjoukon GUESS:in verkkonäkymässä.

6.1.1 Resurssin lisääminen uuteen verkkonäkymään

Selite: verkkonäkymää ei ole käynnistetty, valitut resurssit visualisoidaan uudessa GUESS-ikkunassa.

Painike: show graph

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus (tulosjoukko valittu)	Dogma käynnistää GUESS:in, joka visualisoi resurssit verkossa.
Painallus (ei valittua joukkoa)	-

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Painallus (tyhjä tulosjoukko)	Ilmoitus: resource vector empty

6.1.2 Resurssin lisääminen vanhaan verkkonäkymään

Selite: verkkonäkymä on käynnistetty. Resurssit lisätään jo olemassa olevaan GUESS-ikkunaan.

Painike: add to graph

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus (resurssi, tai resurssijoukko valittu)	Dogma lisää resurssit GUESS:in verkkonäkymään.
Painallus (ei valittuja resursseja)	-

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Painallus (tyhjä tulosjoukko)	Ilmoitus: resource vector empty

6.2 Resurssitaulukko

Selite: Dogma sisältää taulukon, johon käyttä voi hakea resursseja tietokannasta.

6.2.1 Resurssitaulukon järjestäminen attribuuttien mukaisesti

Selite: resurssitaulukon kentät ovat järjestettävissä sarakkeen mukaiseen järjestykseen.

Painike: attribuutin arvo

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus (resurssitaulukon ollessa attribuutin mukaisessa nousevassa järjestyssä)	Resurssitaulukko järjestyy attribuutin mukaiseen laskevaan järjestykseen.
Painallus (resurssitaulukon ollessa attribuutin mukaisessa laskevassa järjestyksessä)	Resurssitaulukko järjestyy attribuutin mukaiseen nousevaan järjestykseen.

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Painallus (resurssitaulukko tyhjä)	-

6.2.2 Resurssitaulukkoon hakeminen tietokannasta

Selite: käyttäjä pystyy hakemaan valitsemiaan hakuehtoja vastaavia resursseja resurssitaulukkoon.

Painike: search

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus (hakuehdot asetettu)	Hakuehtoja vastaavat resurssit tulevat resurssitaulukkoon.

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Resurssikantaa ei ole valittu haun yhteydessä	Ilmoitus käyttäjälle, hakua ei suoriteta
Hakuehdot väärää tyyppiä (esim. luvun kohdalla merkkijono)	Ilmoitus käyttäjälle, hakua ei suoriteta
JDBC-ajuria ei löydy	Käytetään oletusajuria (HSQLDB)
SQL-kyselyssä tapahtuu virhe	Palautetaan null

6.2.3 Lomakenäkymän avaaminen resurssitaulukosta

Selite: käyttäjä pystyy avaamaan yksittäisen resurssin lomakenäkymään muokkausta varten.

Painike: edit

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus	Dogma avaa lomakenäkymän, jonka kentätiedoissa on valitun resurssin attribuutit.

6.3 Resurssin tietojen muokkaus

Selite: käyttäjä pystyy muokkaamaan resurssin attribuutteja lomakenäkymässä.

6.3.1 Resurssin tietojen muokkaus lomakenäkymässä

Painike: apply / commit

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus (attribuutteja muutettu)	Resurssin muutetut tiedot päivittyvät tietokantaan ja resurssitaulukkoon
Painallus (attribuutteja ei muutettu)	-

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Vääräntyyppisiä tietoja kentissä	Ilmoitus käyttäjälle, ei suoriteta muutoksia

6.3.2 Resurssin lisääminen

Selite: käyttäjä pystyy lisäämään uusia resursseja resurssitaulukon ja lomakenäkymän kautta

Painike: add resource (lomakenäkymässä)

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus (vaaditut attribuuttikentät täytetty)	Resurssi lisätään tietokantaan
Painallus (vaadittuja attribuuttikentättä tyhjinä)	Virheilmoitus käyttäjälle

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Ei valittua tietokantaa, johon resurssi lisättäisiin	Luodaan uusi tietokanta

6.4 URI-taulukko

Selite: Resurssilla on URI-taulukko, joka kertoo resurssin sijainnit.

6.4.1 URI:n lisäys

Painike: add URI (valitun resurssin kohdalla)

<i>Syöte</i>	<i>Toivottu tulos</i>
URI	URI lisätään resurssin URI-taulukkoon ja tietokantaan

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
URI:n muoto virheellinen	Ilmoitus käyttäjälle, ei suoriteta lisäystä

6.4.2 URI:n poisto

Painike: remove URI (valitun resurssin kohdalla)

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus	URI poistetaan resurssin URI-taulukosta ja tietokannasta

6.4.3 URI:n avaaminen

Painike: open URI (valitun resurssin kohdalla)

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus	URI avataan tiedostotyyppistä riippuen järjestelmän oletusohjelmalla

6.5 Tapahtumien seuranta ja tallennus

Selite: käyttäjän suorittamat tapahtumat tallennetaan lokitiedostoon.

<i>Tapahtuma</i>	<i>Toivottu tulos</i>
Resurssi esiintyy hakutuloksessa	Tapahtuma kirjataan lokitiedostoon
Resurssi liitetään osajoukkoon	Tapahtuma kirjataan lokitiedostoon
Resurssi avataan lomakenäkymässä	Tapahtuma kirjataan lokitiedostoon
Resurssi avataan oletusohjelmalla	Tapahtuma kirjataan lokitiedostoon
Resurssi liitetään verkkonäkymään	Tapahtuma kirjataan lokitiedostoon
Resurssin attribuutteja muutetaan	Tapahtuma kirjataan lokitiedostoon

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Lisättävä tapahtumamerkkijono null tai 0 merkkiä pitkä	Ei lisätä tapahtumaa lokiin

6.6 Resurssileikepöytä

Selite: resursseja voidaan käsitellä copy/paste -toiminnoilla.

6.6.1 Kopiointi

Painike: copy (yksittäisen resurssin tai osajoukon kohdalla)

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus	Resurssit kopioidaan resurssileikepöydälle.

6.6.2 Liittäminen

Painike: paste (resurssitaulukon kohdalla)

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus	Leikepöydällä olevat resurssit liitetään leikepöydältä valittuun resurssitaulukkoon.

Poikkeukset:

<i>Poikkeus</i>	<i>Käsittely</i>
Kohderesurssitaulukkoa ei ole valittu	-

6.7 Tietokannan valinta

Selite: käyttäjä pystyy valitsemaan mitä tietokantaa käytetään ohjelman toiminnassa.

Painike: select database

<i>Syöte</i>	<i>Toivottu tulos</i>
Painallus	Dogma avaa ikkunan, josta tietokanta voidaan valita.

7. Testattava käyttötapaus

Dogman toimivuus varmistetaan myös asiakkaan toimittaman käyttötapausesimerkin avulla:

Teppo kirjoittaa artikkelia aiheesta "Katsaus PageRank- tyyppisiin solmujen keskeisyysmittoihin". Hän avaa sovelluksen ja tekee vapaatekstihaun "graph centrality pagerank" dokumenttityypisille resursseille. Tulosjoukko sisältää oletusarvoisesti enintään 1000 resurssia ja on järjestetty relevanssin mukaan. Teppo huomaa tulosjoukon alkupäästä selaillessaan muutaman potentiaalisesti mielenkiintoisen artikkelin ja leikkaa ja liittää ne toiseen ikkunaan/tabiin. Sitten Teppo järjestää tulosjoukon siten, että alkuun tulevat hänelle tutuimmat artikkelit ja leikkaa myös osan näistä lähempää tarkastelua varten. Lopuksi Teppo järjestää dokumentit vielä niin, että alussa ovat artikkelit, joita hän ei ole aikaisemmin lukenut ja poimii vielä näistä osan. Sitten Teppo siirtyy lukemaan poimittuja artikkeleita järjestäen ne niiden tuottoajankohdan mukaiseen järjestykseen.

Teppo visualisoi dokumentteihin viittaavat artikkelit verkkoon ja löytää näiden muutamia uusia mielenkiintoisia dokumentteja. Eräs dokumentti sisältää niin hätkähdyttäviä tuloksia, että hän hakee listan dokumentin tekijöistä ja siirtyy hetkeksi heidän kotisivuilleen katselemaan, mitä porukkaa on kyseessä.

Teppo huomaakin erään ennestään tuntemattoman tutkijan, jonka ajatukset näyttävät lupaavilta, ja nostaa tämän toimijan arvoluokitusta parilla pykälällä. Dokumentteja lukiessa vastaan tulee myös dokumentti, jonka sijainnista Tepolla ei ole tietoa. Teppo pyytää sovellusta avaamaan muutaman hakukoneen tätä dokumenttia koskevalla kyselyllä ja löytääkin näin pdf-dokumentin, jonka hän tallentaa sovelluksen pdf-lähdehakemistoon.

Istunnon lopuksi Teppo merkkää kerralla kaikkiin poimittuihin resursseihin tagin "jatka tästä" ja sulkee sovelluksen. Seuraavana päivänä hän jatkaa työtään hakemalla kaikki tagin "jatka tästä" sisältävät resurssit (ja ehkä poistaa tämän tagin). Lisäksi Teppo tekee uudestaan eilisen vapaatekstihaun tällä kertaa lisäehdolla, että imported = tänään. Näiden hän saa listan uusista resursseista, jotka sovellus on yön aikana hakenut tietolähteistä ja indeksoinut hänen käyttöön. Artikkelin kirjoittamisprosessi jatkuu jouhevasti...

On huomattavaa, että tekstihaut esimerkisyöteillä eivät välttämättä tuota tuloksia, joten

ryhmä voi käyttää testauksessa syötteitä, joiden tiedetään olevan resurssitietokannassa.

8. Testausaikataulu

Yksikkötestaus suoritetaan aina luokkien valmistuttua. Integraatiotestaus pyritään aloittamaan 14.11.2006 ja järjestelmätestaus heti integrointitestauksen valmistuttua.