

Testaussuunnitelma

Orava

Helsinki 5.5.2005

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Juhani Bergström

Peter von Etter

Teppo Käsälä

Olli Lyytinen

Jessika Penttinen

Mikko Waris

Asiakas

Eero Hyvönen

Johtoryhmä

Juha Taina

Hannu Räisänen

Kotisivu

<http://www.cs.helsinki.fi/group/orava/>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.2	3.3.2005	Ensimmäinen versio
0.3	11.3.2005	Korjailtu versio
1.0	23.4.2005	Valmis versio

Sisältö

1 Johdanto	1
1.1 Termit	1
1.2 Dokumentin rakenne	1
2 Yleistä	1
2.1 Testauksen kohde ja tavoitteet	1
2.2 Testausympäristö	2
2.3 Testauksen aikataulu	2
3 Testausstrategia	2
4 Testausvaiheet	2
4.1 Yksikkötestaus	2
4.2 Integrointitestausta	3
4.3 Järjestelmätestaus	3
4.4 Hyväksymistestausta	3
5 Testitapausten määrittely	3
5.1 Järjestelmätestit	3
5.2 Xml2rdf-komponentin testaus	4
5.3 Suosittelemääntöjen testaus	5
5.4 Ontologioiden testaus	6
5.5 Käyttöliittymän testaus	6
6 Ominaisuudet, joita ei testata	6
7 Testausraportti	7
Lähteet	7

1 Johdanto

Tämä dokumentti on ohjelmistotuotantoprojekti Oravan testaussuunnitelma. Projektin aiheena on semanttinen videoiden haku- ja suosittelukone. Dokumentissa kuvataan kuinka ohjelmiston testaus suoritetaan. Testauksen tarkoituksena on löytää virheitä ohjelmistosta ja varmistaa että se toteuttaa sille asetetut toiminnalliset ja laadulliset vaatimukset. Lisäksi ohjelmiston toimintojen pitää toimia kuten on suunniteltu ja kaikkien vaatimusdokumentissa esitettyjen prioriteetin 1 toimintojen tulee löytyä ohjelmistosta.

1.1 Termit

LOM lyhenne engl. sanoista *Learning Object Metadata*. Tietomalli opetukseen liittyvien objektien metadatalle. [LOM]

RDF lyhenne engl. sanoista *Resource Description Framework*. Semanttisen web:in tietomalli metadatalle. [RDF]

OntoViews MuseoSuomi-portaalin ohjelmisto. [OntoViews]

1.2 Dokumentin rakenne

Dokumentin luvussa kaksi kerrotaan miksi ja mitä testataan. Käytetty testausstrategia kuvataan luvussa kolme. Luvussa neljä kerrotaan tarkemmin käytettävistä testaus menetelmistä. Luvussa viisi määritellään eri testitapaukset. Testaamatta jätettävät ominaisuudet selvitetään luvussa kuusi. Luku seitsemän kertoo kuinka testauksesta raportoidaan.

2 Yleistä

2.1 Testauksen kohde ja tavoitteet

Testauksen kohteena on semanttinen videoiden haku- ja suosittelukone "Orava". Ohjelmisto on www-palvelu ja se on käytettävissä yleisimmillä internet-selaimilla.

Projektin asiakkaalta on saatu käyttöön Ontoviews-järjestelmä, jota on tarkoitus hyödyntää projektin toteutuksessa. Ohjelmisto koostuu käytännössä kahdesta osasta, XML/RDF-muunnostyökalusta ja Ontoviews-järjestelmästä. Koska Ontoviews-järjestelmä on saatu asiakkaalta, ei sen sisäistä toimintaa testata tämän projektin puitteissa.

Testauksen tavoite on löytää ohjelmistossa mahdollisesti olevat virheet ja puutteet.

2.2 Testausympäristö

Testaus tullaan suorittamaan toteutus/ajoympäristössään. Testauksessa tullaan käyttämään Mozilla ja IE-selaimia. Yksikkötestauksessa käytetään apuna JUnit-työkalua.

2.3 Testauksen aikataulu

Testaus suoritetaan toteutuksen yhteydessä ja projektisuunnitelmassa [PS] testaukseen varattuna aikana. Yksikkötestaus aloitetaan samaan aikaan kuin toteutus. Integroititestausta aloitetaan heti, kun tarvittavat yksiköt on saatu testattua. Järjestelmätestaus aloitetaan integroititestauksen päätyttyä.

3 Testausstrategia

Testaus suoritetaan neljässä osassa: yksikkötestauksena, integraatiotestauksena, järjestelmätestauksena ja hyväksymistestauksena.

Yksikkötestaus suoritetaan toteutusvaiheessa ja se aloitetaan alimmalta tasolta. Yksikkötestauksessa käytetään tarvittaessa apuna tynkiä ja ajureita, simuloimaan muita yksiköitä. Ontoviews-järjestelmän käyttämien ontologioiden ja päättelysääntöjen käyttäytymisen testaus suoritetaan integrointi- ja järjestelmätestausvaiheissa. Järjestelmätestauksessa testataan vaatimusmäärittelyssä esitettyjen vaatimusten toteutumista. Hyväksymistestaus suoritetaan asiakkaalle järjestetyn demotilaisuuden yhteydessä.

4 Testausvaiheet

4.1 Yksikkötestaus

Yksikkötestauksessa keskitytään ohjelman yksikön toimintaan. Testauksella varmistetaan, että kukin yksikkö (luokka) toimii virheettömästi ja kuten sen pitääkin toimia.

Yksikkötestauksessa käytetään lasilaatikkotestausta (white-box) eli testauksessa käytetään hyväksi yksikön lähdekoodia. Lähdekoodin avulla suunnitelluilla testitapauksilla pyritään paikallistamaan mahdolliset virheet yksikön sisällä. Yksikkötestauksessa käytetään tarvittaessa tynkiä ja ajureita, joiden tehtävänä on simuloida muita testauksessa mahdollisesti tarvittavia yksiköitä/moduuleja. Testejä tehdään toteutuksen rinnalla, jolloin aina jonkin yksikön toteutuksen jälkeen sen toiminta testataan. Jokaisen yksikön toiminta pyritään erottamaan muusta koodista ja ympäristöstä niin, että testin läpimeno ei riipu muista ohjelmiston osista.

Yksikkötestaus suoritetaan siten, että jokainen ryhmän jäsen testaa itsenäisesti tekemänsä yksiköt. Yksikkötestauksessa testattavat asiat:

- Rajapinnat

- Tietorakenteet
- Virhetilanteet
- Metodit

Yksikkötestausta ei suoriteta Ontoviews-järjestelmälle, koska sen yksityiskohtiin ei puututa. Jos kuitenkin Ontoviewsiin joudutaan tekemään muutoksia tulee ne testata normaalisti.

4.2 Integroititestausta

Integroititestausta testataan komponenttien välistä kommunikointia. Integroititestausta suorittaa testattavien osien tekijät. Integroititestausta aloitetaan kun integroitavien osien yksikkötestausta on saatu valmiiksi. Integroititestausta suoritetaan alhaalta ylöspäin, aloitetaan toteuttavista kokonaisuuksista ja mennään ylöspäin kohti ohjaavia kokonaisuuksia.

4.3 Järjestelmätestausta

Järjestelmätestausta testataan järjestelmää kokonaisuutena. Järjestelmätestausta pyritään selvittämään että ohjelmisto toimii oikein ja toteuttaa sille asetetut vaatimukset.

Lisäksi testataan erityisesti käyttöliittymän toimintaa ja käytettävyyttä.

4.4 Hyväksymistestausta

Hyväksymistestausta suoritetaan asiakkaan kanssa asiakkaalle järjestettävässä ohjelmiston demo-tilaisuudessa. Jos asiakas haluaa ohjelmistoon muutoksia ja niiden tekoon olisi aikaa, voidaan muutosten toteuttamista harkita. Muissa tapauksissa katsotaan testausta päättyneeksi.

5 Testitapausten määrittely

5.1 Järjestelmätestit

Testataan vaatimusdokumentin käyttötapausten toteutuksen toiminta, testejä suoritetaan useita kertoja eri internet selaimilla, jotta oikeasta toiminnasta voidaan olla varmoja:

KT-1 **Valinnan lisääminen selaamalla:** Valitaan jokin järjestelmän tarjoamista kategorialinkeistä, joilla tarkennetaan hakua. Toistetaan linkkien valintaa.

Oletettu toiminta: Valittu kategoria tarkoittaa hakua ja mahdollistaa alikategorioidensa lisäämisen hakuehtoihin. Muiden saman tason kategorioiden valinta muuttuu mahdottomaksi.

KT-2 **Valinnan lisääminen hakusanalla:** Käytetään hakusanaa, jonka tiedetään olevan aineistossa, ja sanaa, josta tiedetään ettei sellaista ole aineistossa.

Oletettu toiminta: Jos hakusana vastaa kategoriaa, lisätään tämä kategoria tehtyihin valintoihin. Jos hakusana vastaa yksittäistä videota, näytetään kyseisen videon sivu. Jos hakusana vastaa useampaa videota, näytetään linkkilista, josta pääsee videoihin. Jos hakusanalle ei löydy vastaavuutta näytetään siitä ilmoitus ja näkymään ei tule muita muutoksia.

KT-3 **Valinnan poistaminen:** Poistetaan valinta, kun valintoja on tehty yksi ja kun valintoja on tehty useampia.

Oletettu toiminta: Valittu kategoria poistetaan hakuehdoista.

KT-4 **Kaikkien valintojen poistaminen:** Poistetaan kaikki valinnat, kun valintoja on tehty yksi ja kun valintoja on tehty useampia.

Oletettu toiminta: Palataan portaalin perustilaan.

KT-5 **Videon katselu:** Yritetään katsoa KT-1 ja KT-2 valinnoilla löydetty videot.

Oletettu toiminta: Valittu video aukeaa selain ikkunaan.

KT-6 **Palautteen lähettäminen:** Asetetaan palautteen osoitteeksi ryhmän sähköpostilista. Käytetään portaalin palautteenlähetystoimintoa. Tarkistetaan tuliko palaute läpi.

Oletettu toiminta: Käyttäjän järjestelmän asetuksista riippuen linkin painaminen käynnistää sähköpostiohjelman ja antaa viestinkirjoitus lomakkeeseen vastaanottajaksi ryhmän sähköpostilistan.

KT-7 **Järjestelmän käyttöönotto:** Suoritetaan järjestelmän käyttöönottoa, kunnes se on saatu suoritettua viisi kertaa peräkkäin onnistuneesti.

Oletettu toiminta: Järjestelmä on käyttövalmiina.

5.2 Xml2rdf-komponentin testaus

Xml2rdf-komponentti testataan vertaamalla sille annettua syötettä siitä saatuun tulostukseen. Tämä tehdään käsin. Testaus suoritetaan luomalla ainakin 12:sta tiedostosta koostuva testiaineisto ja tarkistamalla, että kunkin tiedoston kohdalla vaadittu tieto löytyy myös xml2rdf:n generoimasta tiedostosta oikeassa muodossa ja ettei mitään ei-vaadittua tietoa löydy tiedostosta.

Olkoon X kaikki validit XML-dokumentit ja $A \subseteq X$ kaikki XML-tiedostot, jotka sisältävät kenttiä tai attribuutteja, joiden sisällöt muunnettaisiin RDF-muotoon. Testiaineistolle T pätee:

- $T = T_1 \cup T_2 \cup T_3$
- $T_1 \cap T_2 \cap T_3 = \emptyset$
- $|T_i| \geq 4, i = 1, 2, 3$
- $T_1 \subseteq A$
- $T_2 \subseteq X \setminus A$
- $T_3 \subseteq X'$

Jotta komponentti läpäisisi testin, on seuraavat asiat tarkistettava:

- Jokaisen joukkoon T_1 kuuluvan tiedoston vaadittujen kenttien ja attribuuttien tiedot löytyvät RDF-tiedostosta oikeassa muodossa.
- RDF-tiedosto sisältää *ainoastaan* joukkoon T_1 kuuluvien tiedostojen sallittuja kenttiä.

5.3 Suosittelesääntöjen testaus

Näkymät muodostavat kategoriasäännöt testataan tarkistamalla seuraavat asiat.

- Kategorioiden juuret määräävät predikaatit toimivat. Tämä testataan havainnoimalla, että portaalin käyttöliittymä esittää käyttäjälle toteutusdokumentissa määritellyt näkymät ja että näiden nimet näkyvät oikein.
- Kunkin juurikategorian alakategoriat määräävät predikaatit toimivat. Tämä testataan havainnoimalla, että kunkin alakategorian määräävä predikaatti tuottaa ainakin jonkin kategorian näkymään ja että ko. kategorian nimet näkyvät oikein.
Mikäli sääntö on monimutkaisempi siten, että se sisältää valinnaisesti suoritettavia osia, valitaan tarkistettavaksi niin monta alakategoriaa, että jokainen kohta predikaatin koodista tulee suoritetuksi ainakin kerran.
- Jokainen lehden määräävä predikaatti toimii. Tämä tarkistetaan samalla tavalla kuin alakategorian määräävän predikaatin toiminta.

Suosittelusäännöt muodostavat predikaatit testataan periaatteessa samalla tavalla kuin edelliset, eli valitsemalla testitapauksina toimivat linkit kohteesta toiseen siten, että kunkin säännön jokainen kohta tulee suoritetuksi ainakin kerran.

5.4 Ontologioiden testaus

Ontologioiden testaus on käsin suoritettavaa työtä, jossa pyritään varmistamaan, että ontologioiden määrittämä käsitemaailma on järkevä. Lisäksi pyritään varmistamaan, että ontologioista on hyötyä Klaffi-aineiston kuvaamisessa.

Ensin testataan ontologioiden järkevyyttä ontologia kerrallaan tutkimalla muodostaako ontologia oman yksikäsitteisen käsitemaailman, joka ei ole ristiriidassa muiden ontologioiden kanssa. Lisäksi mietitään puuttuuko ontologiasta jotain oleellista. Lopuksi vielä pohditaan onko ontologiaa mahdollista hyödyntää Klaffi-aineiston kuvaamisessa.

Teemat-ontologian käsitteet pohjautuvat avainsanoille. Testauksessa katsotaan, että tärkeitä avainsanoja ei ole unohdettu ja varmistetaan, että ontologioissa ei esiinny päällekkäisiä (avainsanoista johdettuja) termejä.

Klaffin metadatan perusteella luodaan testitapaukset jokaiselle merkittävästi erilaiselle kohdevideon metadatalle. Tämä tarkoittaa kohdevideoiden metadatan kenttien eri kombinaatioita. Kaikkia kombinaatioita ei kuitenkaan tarkastella erikseen, vaan kombinaatioista valitaan merkitykselliset, joilla pyritään kattamaan kaikki mahdolliset tilanteet ja erikoistapaukset.

Näin luotuduista testitapauksista tutkitaan käsin soveltuuko video ripustettavaksi käytäviin ontologioihin.

5.5 Käyttöliittymän testaus

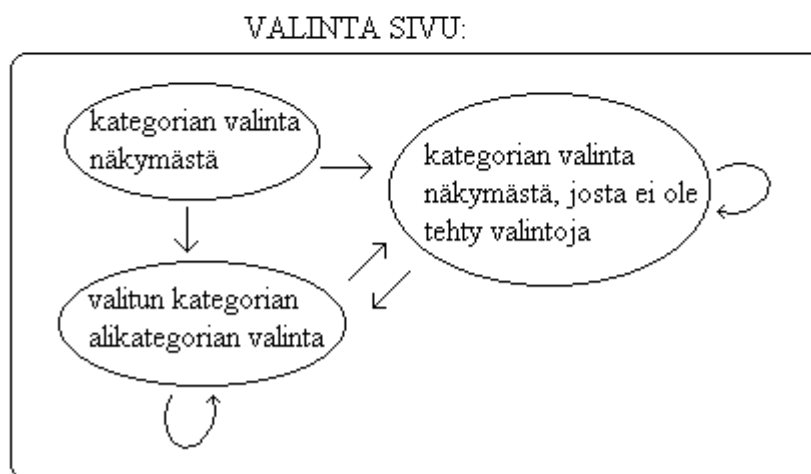
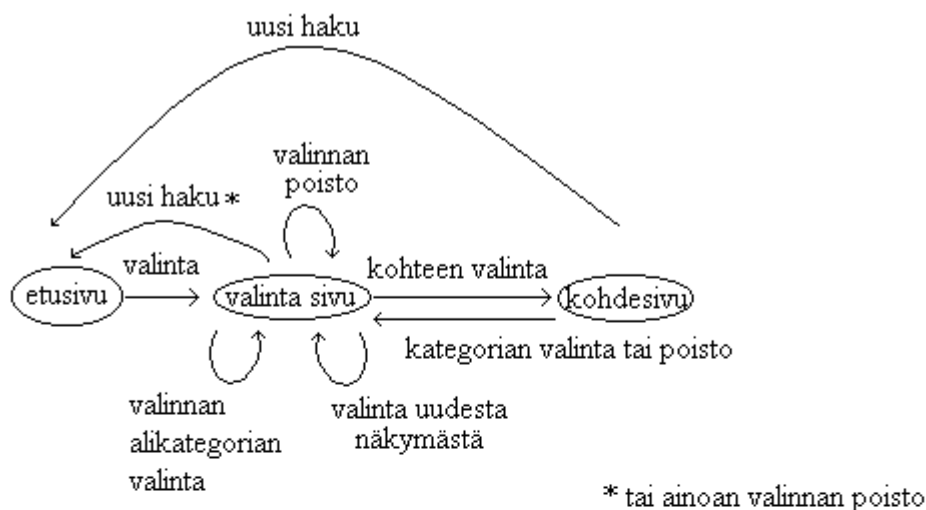
Käyttöliittymän testaus suoritetaan käsin. Testauksella pyritään varmistamaan, että käyttöliittymä mahdollistaa kaikki kuvassa 1 olevat tilasiirtymät ja että niihin reagoidaan oikein.

Palautteen lähettämisen ja ohjeen katsomisen pitää olla mahdollista kaikissa tiloissa (etusivu, valintasivu ja kohdesivu). Uuden haun suorittaminen on mahdollista kaikissa muissa tiloissa paitsi etusivulla, josta uusi haku aina aloitetaan.

Yksittäisistä näkymistä tarkastetaan, että ne eivät mahdollista tyhjien kategorioiden valintaa, eivätkä valitun kategorian tasoisen kategorian tai ylemmän tason kategorian valintaa. Kategorian poistamisen hakuehdoista pitää poistaa myös kaikki poistettavan kategorian alakategoriat.

6 Ominaisuudet, joita ei testata

Ontoviews-paketin sisäisiä toimintoja ei testata, vain sen toiminta käytössä olevalla datalla testataan. Kuitenkin Ontoviewsiin mahdollisesti tehtävät muutokset testataan.



Kuva 1: Käyttöliittymän tilasiirtymäkaavio.

7 Testausraportti

Suoritettujen testien perusteella muodostetaan testausraportti-niminen dokumentti, jossa kuvataan tehdyt testit ja niiden tulokset. Kun jokin testi on saatu valmiiksi, merkitään testausraporttiin sen tulokset.

Lähteet

LOM Learning object metadata. <http://ltsc.ieee.org/wg12/>. [14.2. 2005]

- OntoViews Ontoviews.
<http://www.cs.helsinki.fi/group/seco/museums/dist/>.
[14.2. 2005]
- PS Orava, Projektisuunnitelma.
- RDF Resource description framework. <http://www.w3c.org/RDF/>.
[14.2. 2005]