Kettu-Sovellusprojekti

Sovellusraportti

Henri Koskenranta
Kosti Kuokkanen
Antti Marttila
Terhi Taanonen

Versio: 0.2
Julkinen
14. tammikuuta 2008

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2008		
Tilaaja	__.__.2008		
Ohjaaja	__.__.2008		

Tietoa dokumentista

Tekijät:

- Henri Koskenranta (HK) `heolanko@jyu.fi` 050-3077318
- Kosti Kuokkanen (KK) `koalkuok@jyu.fi` 044-5666304
- Antti Marttila (AM) `antmatma@jyu.fi` 050-3546127
- Terhi Taanonen (TT) `ttaanone@jyu.fi` 050-3624724

Dokumentin nimi: Kettu-projekti, Sovellusraportti
Sivumäärä: 24
Tiedosto: `sovellusraportti.tex`

Tiivistelmä: Tämä dokumentti on sovellusraportti Jyväskylän yliopiston tietotekniikan laitoksen Kettu-sovellusprojektille. Dokumentissa kuvataan Kettu-sovellusprojektin jatkokehittämää sovellusta.
Avainsanat: Kettu, esteettömyys, WCAG, Foxability, ESOK.

Versiohistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.1	19.12.2007	Luonnoksen tekeminen aloitettu.	KK
0.2	8.1.2008	Useista luvuista työversiot. Runko hahmottuu.	KK

Tietoa projektista

Kettu-sovellusprojekti jatkokehitti Firefox-selaimen Foxability-laajennosta ja tuotti siihen WCAG 1.0 suosituksen mukaiset esteettömyystestit.

Tekijät:

- Henri Koskenranta (HK) `heolanko@jyu.fi` 050-3077318
- Kosti Kuokkanen (KK) `koalkuok@jyu.fi` 044-5666304
- Antti Marttila (AM) `antmatma@jyu.fi` 050-3546127
- Terhi Taanonen (TT) `ttaanone@jyu.fi` 050-3624724

Tilaaja:

- Kimmo Aittokallio `kimaitt@jyu.fi` 014-2602746
- Antti Ekonoja `anjoekon@jyu.fi` 014-2602746
- Tommi Lahtonen `tjlahton@jyu.fi` 014-2602746
- Hannu Puupponen `Hannu.Puupponen@adm.jyu.fi` 014-3603734

Ohjaajat:

- Ville Isomöttönen `vilisom@jyu.fi` 0400-608130
- Tarmo Friman `tatafrim@jyu.fi` 044-5815816

Yhteystiedot:

- Sähköpostilistat: `kettu@korppi.jyu.fi`, `kettu_opetus@korppi.jyu.fi`
- Projektiarkisto: `https://korppi.jyu.fi/list-archive/kettu/ind.html`
- Opetusarkisto: `https://korppi.jyu.fi/list-archive/kettu_opetus/ind.html`
- Työhuone: Ag C225.3 / 014-2604971

Sisältö

1 Johdanto 1

2 Käyttöliittymä 2
2.1 Test view . . . 3
2.2 Manage tests . . . 3
2.3 Test results . . . 4
2.4 Käyttöliittymän toteutus . . . 4

3 Rakenne ja toiminnallisuus 6
3.1 Toimintaperiaate . . . 6
3.1.1 Injektointi . . . 7
3.1.2 Sandbox . . . 7
3.2 Rakenne . . . 7

4 Luokat 9
4.1 FoxabilityUI . . . 9
4.2 FA_Category . . . 10
4.3 FA_rdfHandler . . . 10
4.4 FA_Result . . . 10
4.5 FA_TestErrorLogger . . . 10
4.6 FA_TestModule . . . 11
4.7 FA_Test . . . 11
4.8 FA_TestModuleParser . . . 11
4.9 FA_TestSet . . . 12
4.10 FA_ModuleManager . . . 12
4.11 FA_ResultManager . . . 12
4.12 FA_TestRunner . . . 12
4.13 Esteettömyystesti . . . 13

5 Tiedostot ja tietorakenteet 14
5.1 foxability.rdf . . . 14
5.2 XUL . . . 14

6 Testaus ja koodin kehittyminen 16
6.1 Toiminnallisuuksien testaus . . . 16
6.2 Hyväksyntätestaus . . . 16

7 Ohjelmointikäytänteet ja toteutusympäristö **17**
7.1 Lähdekoodin ulkoasu 17
7.2 Toteutusympäristö 17
7.3 Foxabilityn kehitysversion käyttöönotto 18

8 Jatkokehitysideat **19**
8.1 Käyttöliittymä 19
8.1.1 Rivinumerointi 19
8.1.2 Esteettömyys 19
8.1.3 Lokalisointi 19
8.2 Toiminallisuudet 20
8.2.1 EARL 20
8.3 Testit 20
8.3.1 Firefox muokkaa lähdekoodia 20
8.3.2 Sandboxin asettamat rajoitukset domainin ulkopuolle menemisestä 20
8.4 Tiedossa olevat bugit 21
8.4.1 Kategorian poisto 21
8.4.2 Kategorian nimen katoaminen 21
8.5 Toteutusratkaisujen analyysi eli mitä tekisimme toisin 22
8.5.1 Toistuvien funktioiden nosto ylempään luokkaan 22
8.5.2 Yleiskäyttöiset funktiot 22
8.5.3 Käyttöliittymä 22

9 Lähteet **23**

Liitteet

A Termit **24**

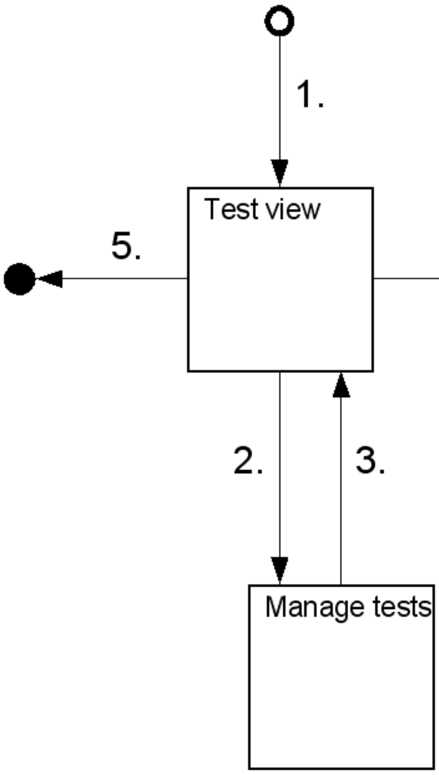
1 Johdanto

Kettu-sovellusprojekti jatkokehitti Foxability-laajennosta Firefox-selaimeen ja tuotti siihen WCAG 1.0 suosituksen mukaiset esteettömyystestit. Foxability-laajennoksen tarkoituksena on automaattinen verkkosivun esteettömyyden tarkistus.

Tämä dokumentti on Kettu-sovellusprojektin sovellusraportti. Sovellusraportissa käydään läpi sovelluksen rakennetta, eri osien toimintaa, sekä toteutusratkaisuja. Sovellusraportin lopuksi esitellään esiin tulleita jatkokehitysideoita.

2 Käyttöliittymä

Luvussa esitellään Foxability-sovelluksen käyttöliittymä ja siitä löytyvät toiminnallisuudet.



Kuva 2.1: Käyttöliittymän ikkunoiden väliset suhteet.

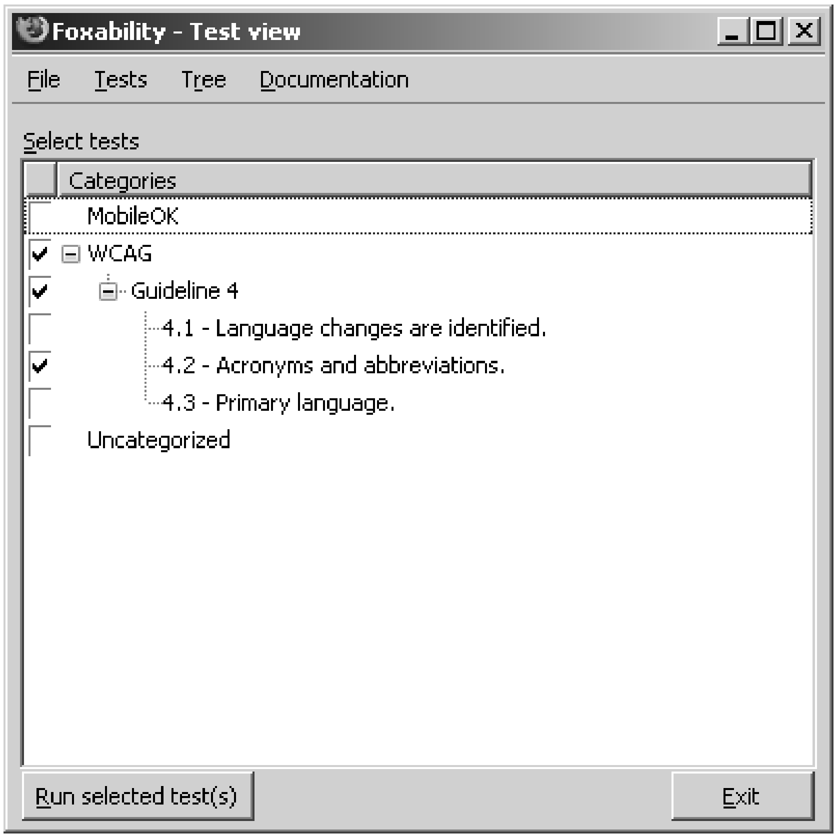
1. Sovellus käynnistetään.

2. Siirrytään *Manage tests* -ikkunaan lisäämään testejä.

3. Lisätyt testit tulevat näkyviin *Test view* -ikkunaan.

4. Valitaan testejä ja suoritetaan ne, jolloin näkymä siirtyy *Test results* -ikkunaan.

5. Suljetaan sovellus.



Kuva 2.2: *Test view* -ikkuna.

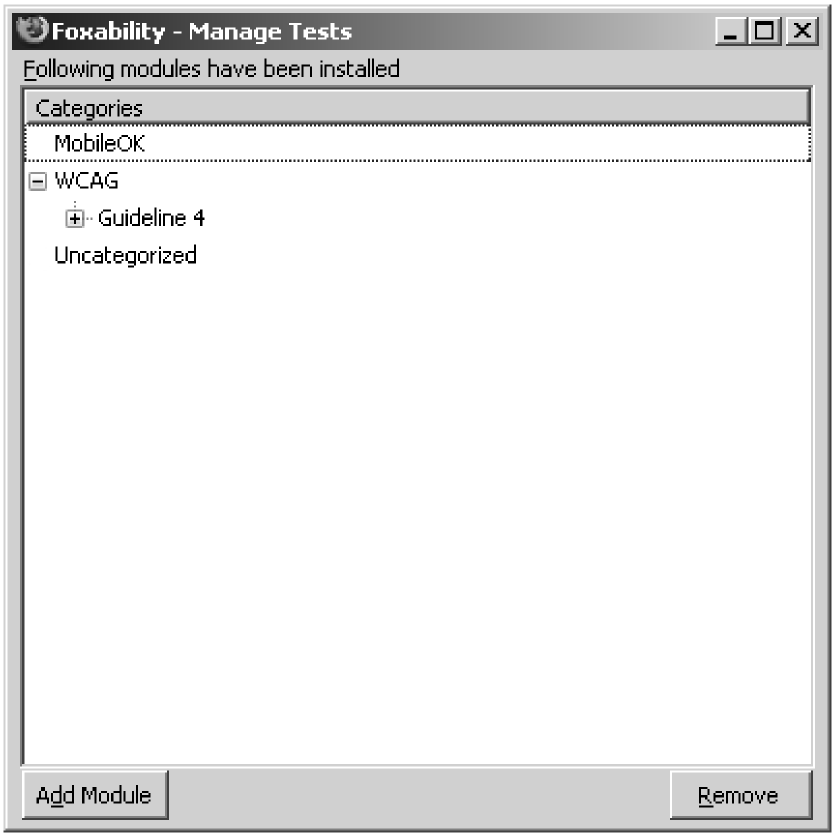
2.1 Test view

Kuvassa 2.2 on *Test view* -ikkuna. Sen tärkeimmät toiminnot ovat testien selaaminen, testien valitseminen ja valittujen testien suorittaminen. Valittujen testien suorittaminen avaa *Test results* -ikkunan. *Test view* -ikkunasta pääsee myös siirtymään *Manage tests* -ikkunaan ja lukemaan käyttöohjetta.

Ikkunassa näkyy luodut kategoriat(esim. WCAG), kategoriassa näkyy siihen asennetut moduulit(esim. Guideline 4) ja moduulissa näkyy siihen kuuluvat testit(esim. 4.3 - Primary language.).

2.2 Manage tests

Kuvassa 2.3 on *Manage tests* -ikkuna. Sen toiminnot ovat moduulien ja kategorioiden hallinta.



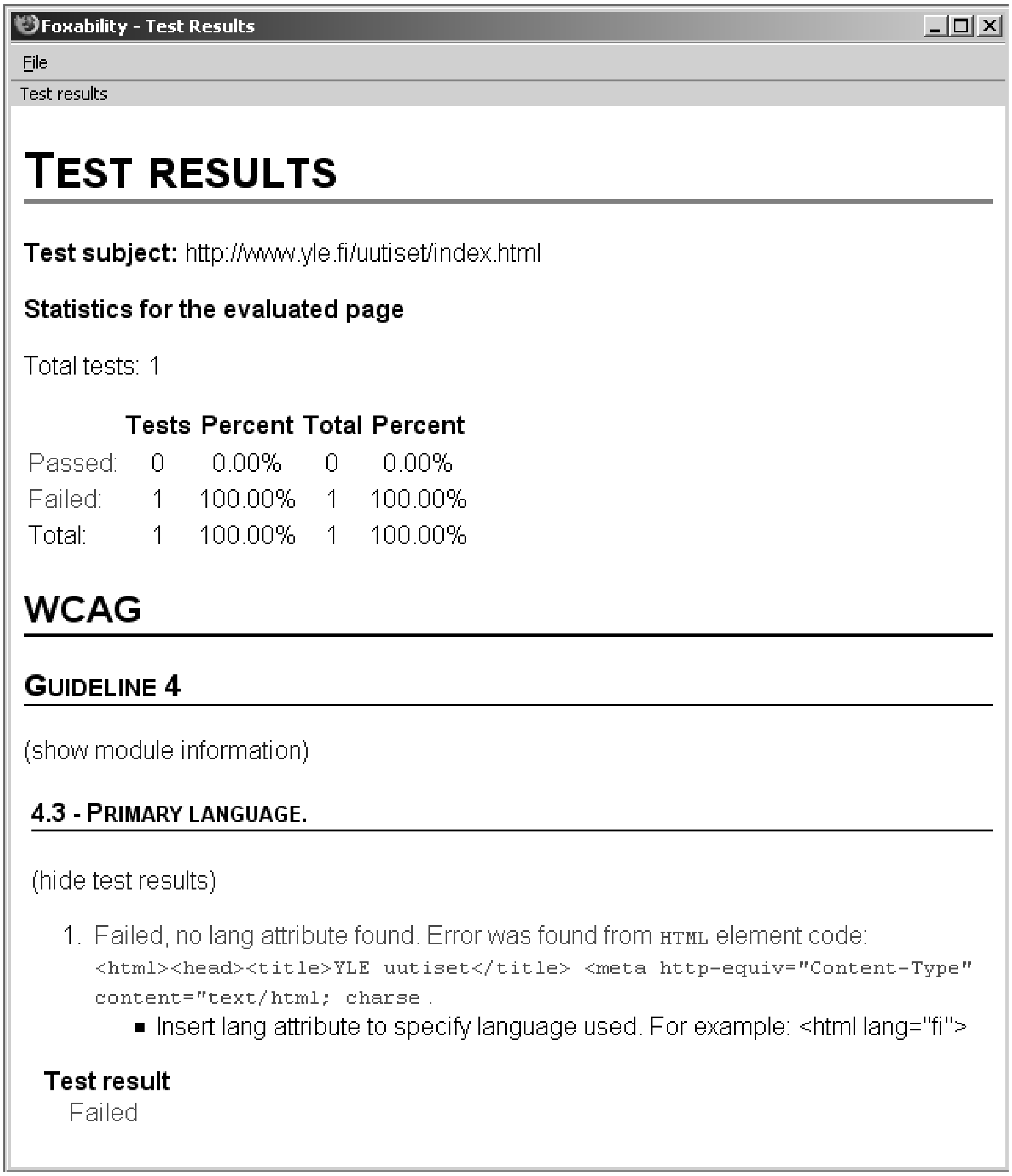
Kuva 2.3: *Manage tests* -ikkuna.

2.3 Test results

Kuvassa 2.4 on *Test results* -ikkuna, jossa näytetään esteettömyystestien tulokset. Lisäksi ikkunassa on tiedot testatun sivun osoitteesta, tilastoja testien tuloksista, testien nimet ja testien tekijöiden yhteystiedot. Testien tulokset ja testien tekijöiden yhteystiedot voi halutessaan pienentää pois näkyvistä.

2.4 Käyttöliittymän toteutus

Käyttöliittymä on toteutettu käyttöliittymän kuvauskieli XUL:lla. XUL on XML-muotoinen, järjestelmäriippumaton merkintäkieli. Esimerkki XUL-tiedostosta on esitetty kuvassa 5.2.

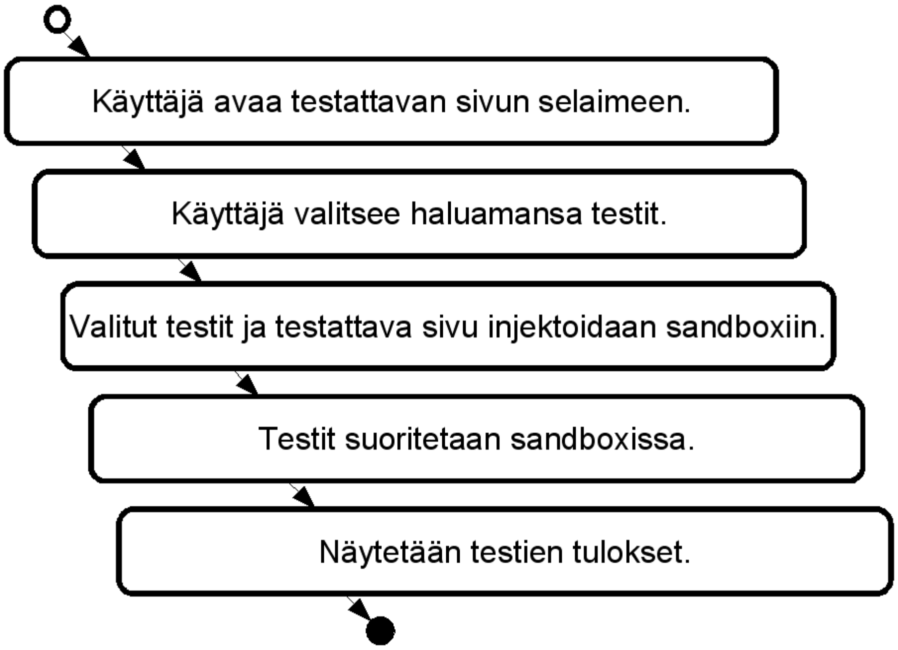


Kuva 2.4: *Test results* -ikkuna.

3 Rakenne ja toiminnallisuus

Tässä luvussa käydään läpi sovelluksen rakennetta, sekä sovelluksen osien toimintaa ja tehtäviä.

3.1 Toimintaperiaate



Kuva 3.1: Foxabilityn yleinen toimintaperiaate.

Foxabilityn tarkoitus on esteettömyystestien ajaminen verkkosivulle ja suoritettujen testien tulosten näyttäminen. Jotta tämä kaikki voitaisiin tehdä, tulee käyttäjän ensin valita testattava sivu. Valinta tapahtuu avaamalla testattava sivu selaimeen ja sen jälkeen siirtymällä Foxabilityyn. Käyttäjän tulee myös valita haluamansa testit, jotka sivulle ajetaan. Valitut testit ja testattava sivu injektoidaan sandboxiin, jossa testit ajetaan. Lopuksi testien tulokset näytetään Foxabilityn avaamassa ikkunassa.

3.1.1 Injektointi

Injektointi tarkoittaa tekniikkaa, jossa ohjelman ulkopuolinen merkkijono lisätään ohjelmaan. Usein merkkijono on lyhyt koodinpätkä ja ohjelma jokin verkkopalvelu. Foxabilityn ulkopuolella, injektointia käytetään usein haittatarkoituksessa. Foxabilityssä injektoidaan valittujen testien lähdekoodit luotuun sandboxiin.

3.1.2 Sandbox

Sandbox on eristetty, tietoturvallinen ympäristö, jossa voidaan suorittaa ohjelmia rajoitetuin oikeuksin. Foxabilityn tapauksessa sandbox takaa käyttäjälle laajennoksen turvallisen käytön, vaikka testi olisi kirjoitettu vahingoittamistarkoituksessa. Rajoitetuista oikeuksista voi olla kuitenkin myös haittaa. Foxability ei pysty tarkistamaan sivuja, jotka ovat käyttäjän valitseman sivun domainin ulkopuolella.

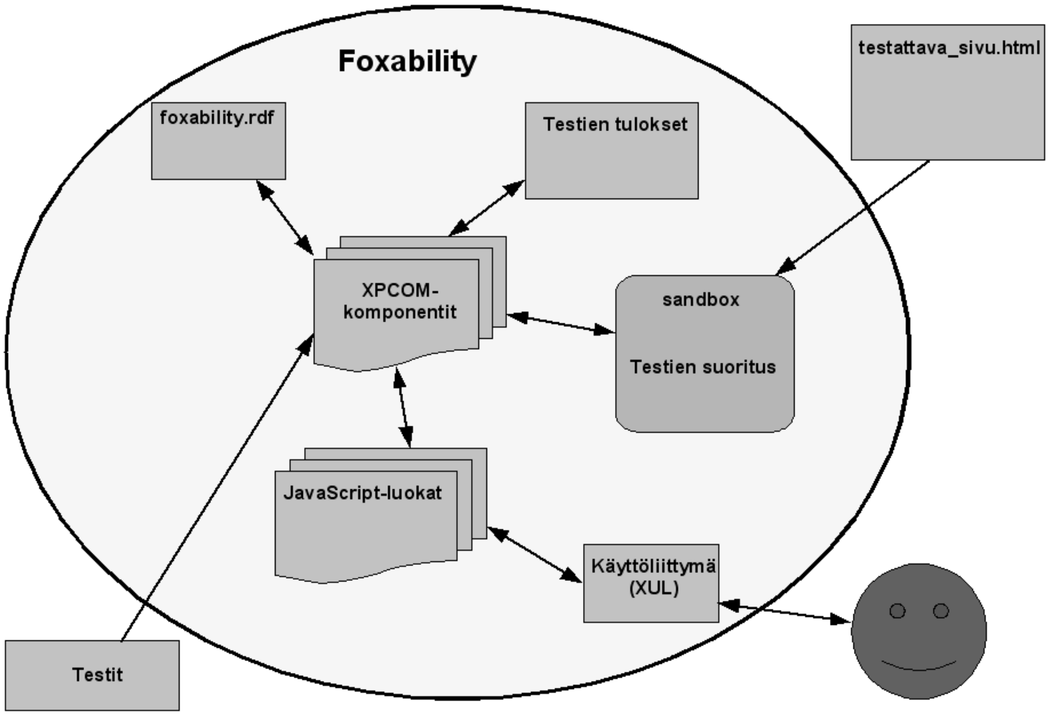
3.2 Rakenne

Kuvassa 3.2 on esitelty Foxabilityn toimintaa ja sen keskeisiä osia. Foxabilityn käyttöliittymä on toteutettu käyttöliittymän kuvauskieli XUL:lla. Käyttöliittymän toteutuksesta lisää kohdassa 5.2. Varsinaiset toiminallisuudet ovat kirjoitettu JavaScriptillä.

Tehtävät, jotka vaativat Geckon toiminnallisuuksien käyttöä, täytyy tehdä XPCOM-komponenteilla. Tällaisia toiminnallisuuksia Foxabilityssä ovat mm. tulosten raportointi, tiedoston käsittely, sekä testien suoritus. XPCOM-komponentit voisi kirjoittaa muillakin kielillä, mutta Foxabilityssä ne on toteutettu JavaScriptillä.

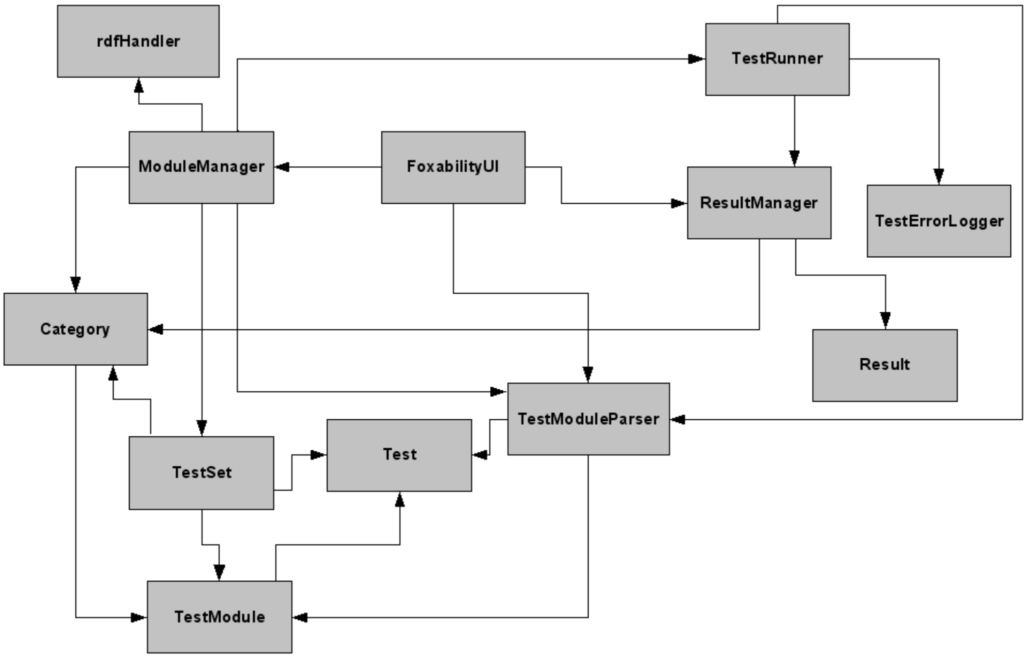
Foxability.rdf tiedostossa pidetään yllä tietoja käyttäjän asentamista testeistä. Itse testit ovat kuitenkin pikemminkin ulkopuolelta tuleva sisältö, kuin valmis osa Foxabilityä.

Varsinainen testien ajo tapahtuu sandboxissa, josta on kerrottu lisää kohdassa 3.1.2



Kuva 3.2: Foxabilityn rakenne.

4 Luokat



Kuva 4.1: Foxabilityn luokkakaavio.

Foxability koostuu yhdeksästä JavaScript-prototyypistä sekä kolmesta XPCOM-komponentista. Kuvassa 4.1 on kuvattu luokkien välisiä yhteyksiä. Luokassa, josta nuoli lähtee, on nuolen kohde joko attribuuttina tai se luodaan lähtevässä luokassa. Prototyypit ja komponentit esitellään tarkemmin alla.

4.1 FoxabilityUI

FoxabilityUI on Foxabilityn käyttöliittymäluokka. Se hoitaa Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.2 FA_Category

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.3 FA_rdfHandler

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.4 FA_Result

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.5 FA_TestErrorLogger

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.6 FA_TestModule

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.7 FA_Test

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.8 FA_TestModuleParser

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.9 FA_TestSet

Parsii käyttöliittymältä saatavan puun perusteella ajettavat testit. Luo ajettaville testeille kategoriat ja moduulit joihin testit ladataan. Nimensäkin perusteella asettaa ajettavat testit. FA_ModuleManager.js tiedostossa oleva FA_ModuleManager prototyyppi käyttää FA_TestSet prototyyppiä apuna ajettavien testien hallinnassa. TestSet taasen käyttää FA_Category.js tiedostossa olevaa FA_Category prototyyppiä apuna kategorian hallintaan ja FA_TestModule.js tiedostossa olevaa FA_TestModule prototyyppiä käytetään moduulien hallintaan.

4.10 FA_ModuleManager

Käpistelee FA_rdfHandler.js tiedostossa olevan FA_rdfHandler prototyypin kautta rdf-tiedostoa, joka toimii ohjelman tietokantana. Nimensä mukaan ModuleManager hoitaa moduulien ja sitä kautta testien päivityksen rdf-tiedostoon/tiedostosta. Hoitaa FA_TestSet.js tiedostossa olevan FA_TestSet prototyypin luoman ajopuun eteenpäin FA_TestRunnerille ajettavaksi.

4.11 FA_ResultManager

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.12 FA_TestRunner

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean rhoncus. Donec felis. Nulla in mauris nec augue adipiscing mollis. Morbi elit. Duis ac nisi. Ut ornare metus. Quisque rutrum placerat justo. Duis lorem est, cursus vitae, euismod non, tincidunt vel, dolor. Fusce volutpat dui eu risus. Quisque nibh neque, placerat ut, laoreet vitae, fermentum non, sapien. Maecenas malesuada lorem at tellus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

4.13 Esteettömyystesti

Näihin tiedostoihin on kirjoitettu varsinaiset testit javascriptillä ja apuna on käytetty DOM-rajapintaa dom-puun käsittelyyn, joka siis sisältää tutkittavan nettisivun helposti käsiteltävässä muodossa. Koska Esteettömyystesti-tiedostojen sisältämät koodit on injektoitu nettisivuun kuin ne olisivat siellä olevia scriptejä, tarvitaan scripteistä yhteys itse Foxabilityyn ja sitä varten scriptissä käytettävät koodit täytyy periä TestModule.js tiedostossa sijaitsevasta FA_TestModule prototyypistä. Näin testien tulokset saadaan siirrettyä hiekkalaatikosta itse Foxabilitylle.

5 Tiedostot ja tietorakenteet

Tässä luvussa kuvataan ohjelman käyttämien tiedostojen rakenteita ja tarkoituksia.

5.1 foxability.rdf

```
<?xml version="1.0"?>
<RDF:RDF xmlns:NS1="http://foxability.sourceforge.net/"
         xmlns:NC="http://home.netscape.com/NC-rdf#"
         xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <RDF:Seq RDF:about="http://foxability.sourceforge.net/MobileOK">
  </RDF:Seq>
  <RDF:Description RDF:about="http://foxability.sourceforge.net/MobileOK"
                   NS1:categoryName="MobileOK" />
  <RDF:Seq RDF:about="http://foxability.sourceforge.net/allCategories">
    <RDF:li RDF:resource="http://foxability.sourceforge.net/MobileOK"/>
    <RDF:li RDF:resource="http://foxability.sourceforge.net/WCAG"/>
    <RDF:li RDF:resource="http://foxability.sourceforge.net/Uncategorized"/>
  </RDF:Seq>
  <RDF:Seq RDF:about="http://foxability.sourceforge.net/WCAG">
  </RDF:Seq>
  <RDF:Description RDF:about="http://foxability.sourceforge.net/WCAG"
                   NS1:categoryName="WCAG" />
  <RDF:Seq RDF:about="http://foxability.sourceforge.net/Uncategorized">
  </RDF:Seq>
  <RDF:Description RDF:about="http://foxability.sourceforge.net/Uncategorized"
                   NS1:categoryName="Uncategorized" />
</RDF:RDF>
```

Kuva 5.1: Foxability.rdf.

Käyttäjä voi asentaa ja poistaa testejä Foxabilityyn. Jotta Foxability muistaisi asennetut testit käynnistyskertojen välissä, pitää tieto niistä olla tallessa jossakin. Tieto asennetuista testeistä ylläpidetään foxability.rdf tiedostossa. Rdf-tiedosto on XML-muotoinen tiedontallennusstandardi. Kuvassa 5.1 on foxability.rdf:n sisältö ja siinä näkyy asennuksen yhteydessä luotavat kolme kategoriaa.

5.2 XUL

XUL on XML-muotoinen, järjestelmäriippumaton, käyttöliittymän kuvauskieli. Kuvauskielellä muotoillaan käyttöliittymän ulkoasu eli käyttöliittymäkomponenttien

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE overlay SYSTEM "chrome://foxability/locale/foxability.dtd">

<overlay
  id="faOverlay"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <script type="application/x-javascript" src="fa.js"/>
  <menupopup id="menu_ToolsPopup">
    <menuitem label="&overlay.description;" insertafter="devToolsSeparator"
      oncommand="FoxabilityUI.faMain_open();" accesskey="&overlay.descriptionAccesskey;"/>
  </menupopup>
</overlay>
```

Kuva 5.2: Esimerkki XUL:sta.

ulkomuodot ja sijainnit. Lisäksi kuvauskielellä osoitetaan käyttöliittymäkomponenttien tapahtumankäsittelijöille kutsuttavat aliohjelmat. Kuvauskieltä on käytetty Foxabilityn kaikkien seitsemän eri näkymän luomiseen. Kuvassa 5.2 on tiedosto faOverlay.xul kokonaisuudessaan.

6 Testaus ja koodin kehittyminen

Testauksen tarkoituksena on varmistaa, että sovellus täyttää sille asetetut niin toiminnalliset, kuin laadulliset vaatimuksetkin. Seuraavaksi käydään lyhyesti lävitse kuinka testaus käytännössä toteutui.

6.1 Toiminnallisuuksien testaus

Ohjelmointitehtävät Kettu-sovellusprojektissa jakautuvat selvästi kahtia: esteettömyystestien toteuttamiseen ja käyttöliittymän parantamiseen sekä bugien korjaukseen.

Esteettömyystestien kohdalla testin toteuttaja kirjoitti itse testisivuja, joita vastaan esteettömyystestien toimivuutta testattiin. Testisivut pohjautuivat tilaajan kanssa tehtyihin tulkintoihin WCAG 1.0 suosituksesta. Käyttöliittymään liittyvien ohjelmointitöiden testaus tapahtui kehitettyjen toiminallisuuksien monipuolisella käyttämisellä.

6.2 Hyväksyntätestaus

Hyväksyntätestausta tapahtui sekä iteraatioiden välisissä palavereissa, että vapaamuotoisemmin projektiryhmän työhuoneessa. Hyväksyntätestaus oli käytännössä sitä, että esteettömyystestin tai käyttöliittymän toimintaa esiteltiin tilaajalle ja tilaaja antoi siitä palautetta. Lopuksi sovelluksen osa joko hyväksyttiin, hyväksyttiin muutoksin, hyväksyttiin kieliasua lukuunottamatta tai ei hyväksytty ollenkaan. Jos hyväksyntätestaus tapahtui vapaamuotoisessa tilaisuudessa projektiryhmän työhuoneessa, merkittiin mahdollinen hyväksyminen seuraavan palaverin pöytäkirjaan.

7 Ohjelmointikäytänteet ja toteutusympäristö

7.1 Lähdekoodin ulkoasu

```
/**
 * CommentModel
 * Comment Model class doesn't do anything. [jargon]
 *
 * @author kettu http://sovellusprojektit.it.jyu.fi/kettu/
 * @author mina <mina@jyu.fi>
 *
 * @version x.x
 * @extends FA_TestModule
 *
 */

/**
 * Comment model
 * @constructor
 */
function CommentModel() {
  this.name = FA_TestModule.name;
}

CommentModel.prototype = new FA_TestModule;

  */
  getTestByName: function(testName) {
    return this.getTestIndexByName(testName);
  }
};
```

Kuva 7.1: Esimerkki lähdekoodin muotoilukäytänteestä.

Lähdekoodin muotoilussa ja kommentoinnissa jatkettiin samoilla linjoilla, mitä Foxability-projektissa oli käytetty. Eclipse-ohjelmointiympäristö aiheutti yllättäviä ongelmia koodin yhtenäisen muotoilun kanssa. Luokkien ja metodien kommentointi on tehty yllä olevan, javadocmaisen, muotoilukäytänteen mukaan. Lähdekoodi kirjoitettiin englanniksi.

7.2 Toteutusympäristö

Foxability-sovelluksen jatkokehitys oli luonnollista tehdä samalla kielellä kuin Foxabilityä oli siihen asti tehty, eli JavaScriptillä. Toteutus tehtiin Eclipse 3.2 -ohjelmointiympäristössä. Eclipsessä käytettiin lisäksi kahta laajennosta: Subclipse ja JSEclipse. Subclipse mahdollisti versiohallinan käytön suoraan Eclipsen kautta ja JSEclipse sai ohjelmointiympäristön ymmärtämään JavaScriptiä.

Firefoxissa käytimme myös kahta laajennosta: Extension Developer Extension ja Web Developer Extension. Extension Developer Extensionilla saa nopeasti työn alla olevan laajennoksen testikäyttöön ja Web Developer Extension on verkkosivujen teossa käytetty yleinen työkalu.

7.3 Foxabilityn kehitysversion käyttöönotto

Tässä luvussa neuvotaan, miten Foxabilityn kehitysversion saa asennettua käyttöön.

1. Hae uusin versio Foxabilitystä. [1]
2. On suositeltavaa luoda oma profiili Firefoxiin, jolle Foxability ym asennetaan. Luo sellainen käynnistämällä Firefox parametrin -p avulla. Start -> Run -> (esim. C:\polkuFirefoxiin\firefox -p)
3. Hae ja asenna Extension Developer Extension. [2]
4. Käynnistä Extension Developer Extension ja valitse työhakemistoksi Foxabilityn extension-kansio.
5. Paina install for development. Käynnistä Firefox uudestaan. Nyt Firefoxin tools-valikkoon on ilmestynyt Foxability Evaluator -valinta ja sitä painamalla Foxability käynnistyy.

8 Jatkokehitysideat

Luvussa käsitellään laajennoksen kehittämisideoita, joita tämän projektin puitteissa ei pystytty toteuttamaan. Lisäksi pohditaan, mitä asioita olisi kannattanut jälkikäteen ajateltuna tehdä toisin.

8.1 Käyttöliittymä

8.1.1 Rivinumerointi

Ehkä tärkein yksittäinen ominaisuus, joka Foxabilityyn tulisi kehittää, on esteettömyystestin löytämän virheen sijainnin rivinumeron näyttäminen. Rivinumeron näyttäminen helpottaisi virheellisten kohtien löytämistä lähdekoodista. Rivinumeroinnin toteuttamisen eteen tehtiin tutkimustyötä, mutta resursseja sen toteuttamiseen ei Kettu-sovellusprojektilla ollut. Rivinumeroinnin implementointi jälkikäteen on hyvin työlästä ja sen helppo toteutus olisi vaatinut asian huomioonottamista jo Foxability-projektin suunnitteluvaiheessa. Lisäksi rivinumerointiin liittyy kohdassa 8.3.1 esitelty ongelma.

8.1.2 Esteettömyys

Esteetön esteettömyystarkistin on periaatteellinen vaatimus, jonka toteuttamiseen Kettu-sovellusprojektin yhteydessä ei osoitettu resursseja. Onko asiallista huomautella muiden esteellisyydestä, kun itse huomautteleva ohjelmakin on esteellinen?

8.1.3 Lokalisointi

Yksi Kettu-sovellusprojektin päämääristä oli saada laajennos helposti lokalisoitavaksi. Lokalisoida tulisi niin käyttöliittymä kuin myös testit. Sitä silmällä pitäen testeissä merkkijonot ovat kirjoitettu kootusti. Käyttöliittymän lokalisointi on pääosin vain kääntämistyötä. Käyttöliittymässä poikkeuksen tekee tulosikkunat, joissa työn määrä voi olla suurempi. Arvioimme käyttöliittymän ja testien lokalisoimiseen kuluvan n. XX h per lokalisoitava kieli.

8.2 Toiminallisuudet

8.2.1 EARL

Evaluation and Report Language eli EARL, on testien tuloksien kuvaamiseen tehty kieli. Jo Foxability-projektin ajoilta juontava tavoite on saada Foxability tallentamaan testien tulokset EARL-muotoon.

8.3 Testit

8.3.1 Firefox muokkaa lähdekoodia

Firefox on niin kettu selain, että verkkosivut näytetään muokkattuun lähdekoodiin perustuen. Lähdekoodin muokkaus voi olla harmitonta lopetuselementtien lisäämistä, tai harmillista elementtien keskinäisten paikkojen vaihtelua (???). Lähdekoodia muokataan ilmeisesti sen takia, että DOM-puun rakentaminen vaatii eheän lähdekoodin.

Kun Foxabilityllä tarkistetaan verkkosivuja saa Foxability käyttöönsä tämän muokatun lähdekoodin perusteella rakennetun DOM-puun. Kun testien raportit perustuu DOM-puuhun, voi virheilmoituksissa näkyä lähdekoodin pätkiä, joita alkuperäisessä lähdekoodissa ei ole.

Kohtaan 8.1.1 liittyen, muokattuun lähdekoodiin perustuva rivinumerointi antaisi rivinumerot mahdollisesti väärin.

Jatkokehitysideana testit olisi ehkä syytä perustua alkuperäiseen lähdekoodiin. (Tässä voisi ehkä vielä kertoa, miten siihen alkuperäiseen lähdekoodiin kuitenkin on mahdollisuus päästä käsiksi. Ja sitten, vaikka siihen pääsee käsiksi, niin ongelmat ei poistu, vaan lähinnä muuttaa muotoaan.)

8.3.2 Sandboxin asettamat rajoitukset domainin ulkopuolle menemisestä

Sandboxin rajoituksista johtuen, Foxability ei pysty käsittelemään tiedostoja, jotka sijaitsevat testattavan sivun domainin ulkopuolella. Koska domainin ulkopuoliset tiedostot (esim. CSS) eivät ole harvinaisia, olisi kätevää, jos niitäkin pystyttäisiin käsittelemään.

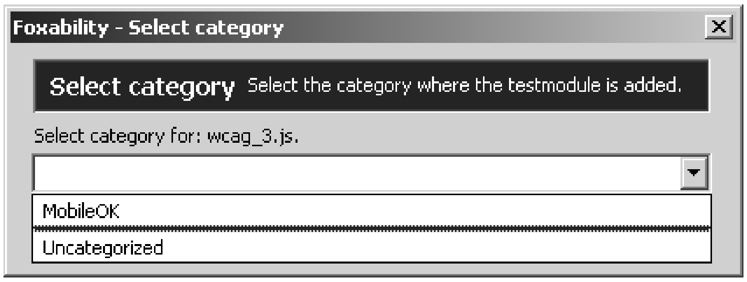
8.4 Tiedossa olevat bugit

8.4.1 Kategorian poisto

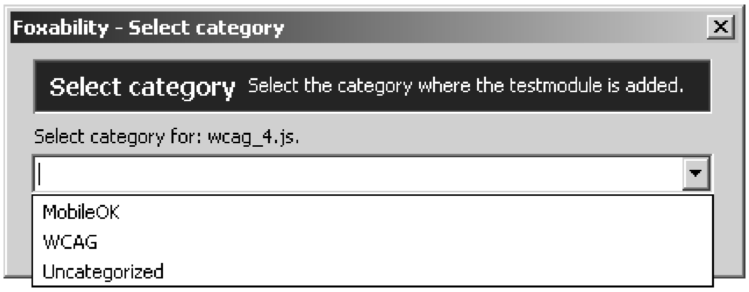
Kategorian poisto *Manage tests* -ikkunassa ei toimi. Ongelma liittynee rdf-tiedoston käsittelyyn(?).

8.4.2 Kategorian nimen katoaminen

Moduuleja lisättäessä kategorioihin esiintyy seuraava bugi: Lisättäessä moduuli kategoriaan, jossa on ennestään vähintään yksi moduuli, ei kategorian nimeä näy valittavien kategorioiden listassa. Kategorian voi kuitenkin valita, mutta se näkyy vain viivana valittavien kategorioiden listassa. Kuvat 8.1 ja 8.1 havainnollistavat virheen ilmenemistä.



Kuva 8.1: Yhden kategorian nimi näkyy vain ohuena viivana.



Kuva 8.2: Kategorian nimet näkyy oikein.

8.5 Toteutusratkaisujen analyysi eli mitä tekisimme toisin

8.5.1 Toistuvien funktioiden nosto ylempään luokkaan

Useassa testissä käytetään samoja aliohjelmia, joten sellaiset aliohjelmat olisi hyvä nostaa FA_TestModuleen, josta testit peritään. Tällaisia aliohjelmia ovat niin testitulosten raportointiin liittyvät kuin myös useassa testissä tarvittavat aliohjelmat.

8.5.2 Yleiskäyttöiset funktiot

Joidenkin testien yhteydessä on hyödynnetty toiseen testiin tarkoitettuja aliohjelmia. Tästä on voinut seurata epätäsmällinen virheilmoitus. Virheilmoituksen olisi syytä olla aina mahdollisimman täsmällinen ja perusteleva. Toisin sanoen aliohjelmista pitäisi tehdä mahdollisuuksien mukaan yleiskäyttöisiä, eikä testikohtaisia.

8.5.3 Käyttöliittymä

Lorem ipsum.

9 Lähteet

[1] Foxability-laajennos: http://foxability.sourceforge.net/

[2] Extension Developer Extension -laajennos Firefoxiin: http://ted.mielczarek.org/code/mozilla/extensiondev/

A Termit

esteettömyys	dasfdas
EARL	Evaluation and Report Language
termi	Lorem ipsum.
XPCOM	Cross Platform Object Model.
sandbox	eristetty, tietoturvallinen ympäristö, jossa voidaan suorittaa ohjelmia rajoitetuin oikeuksin.
injektointi (sandboxiin)	Lorem ipsum.
XUL	XML User Interface Language. Käyttöliittymän kuvauskieli.
rdf	Resource Description Framework.