

Semantic Web – Metadata editor

Ohjelmistotuotantoprojekti, kesä 2002

Ohjelmistotuotantoryhmä 1, Meedio

<http://www.cs.Helsinki.FI/group/meedio>

Mikko Apiola (M.A.)

Ari Inkovaara (A.I.)

Miikka Junnila (M.J.)

Justus Karekallas (J.K.)

Pekko Parikka (P.P.)

Helsinki 10. heinäkuuta 2002

Määrittelydokumentti

Helsingin yliopisto

Tietojenkäsittelytieteen laitos

Versiohistoria

Versio	Pvm	Laatija	Kommentti
0.1	18.6.2002	Pekko Parikka	Alustava versio kommentoitavaksi
0.2	18.6.2002	Miikka Junnila	Kuvaus käyttöliittymästä –luku lisätty
0.3	19.6.2002	Pekko Parikka	Toimintojen määrittelyjä tarkennettu, kuvaus käyttöliittymästä –lukua muutettu.
0.4	19.6.2002	Pekko Parikka	Dokumentin rakennetta muutettu uusien vaatimusten mukaiseksi, toimintoja ja ohjelman rakennekuvausta muutettu ohjelman uusien vaatimusten mukaiseksi, uusia toimintoja lisätty, kielellistä ulkoasua hiottu
0.5	24.6.2002	Justus Karekallas	Dokumentin oikeinkirjoitus tarkistettu ja korjattu
0.6	26.6.2002	Mikko Apiola	Dokumentti muokattu FTR-kokouksen vaatimusten mukaiseksi
0.7	28.6.2002	Mikko Apiola	Dokumenttia viimeistely
1.0	10.7.2002	Mikko Apiola	Virheet korjattu

Sisältö

1	JOHDANTO (P.P.)	1
1.1	Järjestelmän tarkoitus	1
1.2	Tuote	1
1.3	Yleiskatsaus dokumenttiin	1
2	YLEISKUVAUS (M.J.)	2
2.1	Yleinen toiminta	2
2.2	Toimintaympäristö	2
2.3	Käyttäjät	2
2.4	Oletukset ja riippuvuudet	3
3	OHJELMISTOLLE ASETETUT VAATIMUKSET (M.A.)	3
3.1	Järjestelmän tarjoamat palvelut	3
3.2	Lisäominaisuuksiin varautuminen	4
4	TOIMINTOKUVAUS (M.A.)	5
4.1	Sidosryhmät	5
4.2	Järjestelmän yleisarkkitehtuuri	5
4.3	Järjestelmän toiminnot	6
4.3.1	Konfiguraatiodiedoston käsittely	6
4.3.2	XML-validaattori	8
4.3.3	RDF-käsittelijä	8
4.3.4	Tiedostonkäsittelijä	8
4.3.5	Käyttöliittymä	9
4.3.6	Tietoliikenne-osio	10
4.4	Järjestelmän toimintojen tärkeysjärjestys	10
4.5	Pakollisten toimintojen määrittely	12
4.6	Lisätoimintojen määrittely	17
5	KUVAUS KÄYTTÖLIITTYMÄSTÄ (M.J. & J.K.)	19
5.1	Käyttäjärühmät	20

5.1.1	Museohenkilö	20
5.1.2	Ylläpitäjä	20
5.2	Käyttötapaukset	20
5.3	Toteutettavat osat	21
5.4	Toteutustekniikat	22
5.5	Erityisvaatimukset	22
6	MUUT OMINAISUUDET (J.K.)	23
6.1	Ulkoiset liittymät	23
6.2	Ylläpidettävyys	23
6.3	Siirrettävyys, yhteensopivuus	23
7	TESTAUS (J.K.)	23
7.1	Järjestelmätestaus	24
7.2	Muut testausvaiheet	24
8	VIITTAUKSET LÄHTEISIIN	24

Liitteet

Liite 1: Kuva käyttöliittymästä

1 Johdanto (P.P.)

Tässä dokumentissa määritellään ohjelmisto, joka toteutetaan Helsingin yliopiston Tietojenkäsittelytieteen laitoksen (TKTL) ohjelmistotuotantoprojektissa 28.5. - 30.8.2001.

Toteutettava ohjelmisto on museotyöntekijän työkalu, jonka avulla XML-muodossa olevaan museon esinetietoon voidaan liittää semanttisia kuvauksia.

Ohjelmistotuotantoprojektiryhmän jäsenet ovat Mikko Apiola, Ari Inkovaara, Miikka Junnila, Justus Karekallas ja Pekko Parikka. Projektipäällikkönä toimii Pekko Parikka ja sihteerinä Mikko Apiola.

Asiakkaana on prof. Eero Hyvönen TKTL:lta. Ohjaajana toimii Vilho Raatikka ja projektin vastuuhenkilönä Turjo Tuohiniemi.

1.1 Järjestelmän tarkoitus

Työ liittyy tietojenkäsittelytieteen laitoksella käynnissä olevaan älykkäiden, Semantic Web -teknologioihin perustuvien tiedonhaun ja -hallinnan välineiden tuottamiseen museokokoelmien WWW-käyttöön liittyen (Kansallismuseo, Espoon kaupungin museo, Helsingin yliopiston museo).

Ohjelmalla luetaan tietokannasta tuotettu XML(kts luku 8)-tiedosto, jossa on museon esineet kortteina. Ohjelma validoi XML-tiedoston halutun XML-skeeman mukaisesti ja näyttää virheellisesti täytetyt kentät korostettuina. Tämän lisäksi ohjelmalla voidaan XML-korttiin liittää semanttista tietoa esineestä halutun ontologian mukaisesti.

Toteutettavan järjestelmä liittyy Finnish Museums Online (FMO) – hankkeeseen ja sen tarkoituksena on helpottaa museotyöntekijän työtä hänen muuntaessaan esinetietoja XML- ja RDF(kts. luku 8)-muotoon ja lisätessä niihin semanttista tietoa. Tämän lisäksi ohjelman on tarkoitus olla apuna minkä tahansa semanttisen XML-tiedon ja RDF-dokumenttien tekemisessä, esim. keltaiset sivut-hankkeen käytössä.

1.2 Tuote

Tuote on toimiva editoriohjelma, joka tarjoaa mahdollisuuden validoida XML-tiedoston ja lisätä siihen RDF-tietoa. Editoriohjelma on tarkoitettu museotyöntekijälle, joka käsittelee museon esinetietokannasta muodostetun XML-tiedoston sisältämiä esinekortteja ja hän muokkaa esineisiin liittyvät tiedot oikeamuotoisiksi ja lisää esineisiin museon esinetietojen ontologian mukaisen luokittelun. Ohjelmassa varaudutaan myös siihen, että se tulee tulevaisuudessa muidenkin, esim. keltaiset sivut-hankkeen käyttöön.

1.3 Yleiskatsaus dokumenttiin

Järjestelmän tausta ja tavoite, sekä tämän dokumentin rakenne kerrotaan ensimmäisessä luvussa. Toisessa luvussa on käsitelty toimintaympäristö ja käyttäjät. Toteutuksen rajaukset on kuvailtu kolmannessa luvussa. Neljännen luvun toimintokuvauksessa esitellään toteutettavan järjestelmän päätoiminnot. Viidennessä luvussa kuvataan ohjelman käyttöliittymä ja käyttäjäryhmät.

Kuudennessa luvussa kuvataan ohjelmiston ulkoiset liittymät ja kerrotaan niistä ominaisuuksista, jotka eivät ole toiminnallisia. Seitsemännessä luvussa kuvataan toteutuksen aikana ja toteutuksen jälkeen ohjelmistoon sovellettavat testausmenetelmät.

2 Yleiskuvaus (M.J.)

Tämä luku kuvaa ohjelmiston yleistä toimintaa ja tulevan käyttöympäristön sekä tuotettavan ohjelmiston käyttäjät ja toteutukseen liittyviä rajoituksia.

2.1 Yleinen toiminta

Ennen varsinaista ohjelman käynnistämistä määritellään metadataeditorin konfiguraatitiedostoon (esim. kts. luku 4.3.1) käytettävä RDF(S)(kts luku 8)-ontologia ja XML-skeema(kts. luku 8). Sinne syötetään myös tagit, jotka XML-tiedostoissa rajaavat yhden ”XML-kortin”, joka kerrallaan tulee käsiteltäväksi.

Käynnistettyään ohjelman käyttäjä voi avata XML-tiedostoja ja ottaa niistä käsiteltäväksi yksittäisiä kortteja. Ohjelma validoi valitun XML-kortin määritellyn XML-skeeman mukaisesti ja tulostaa kortin tiedot esiin. Virheellisten kenttien taustaväri on punainen ja ohjelma tarjoaa korjausohjeita, mikäli ne on sisällytetty XML-skeemaan. Käyttäjä voi korjata virheelliset tiedot sekä päivittää muita tietoja. Kortin kenttiin saattaa liittyä muuta niihin liittyvää tietoa erityisattribuutteina, jotka määritellään konfiguraatitiedostossa. Esim. kenttien/kortin julkaisukelpoisuusattribuutti. Skeeman mukaan saattaa kortin tietoihin liittyä myös muita kenttiä, esimerkiksi ontologiaan toivottavat muutokset.

XML-validoinnin lisäksi korttiin liitetään konfiguraatitiedostossa määritellyn RDF-skeeman mukaista RDF-tietoa. Ohjelma ehdottaa valmiiksi tiettyjä luokkia esineelle, mikäli XML-kortin tiedot vastaavat RDF-Skeeman luokkien nimiä. Käyttäjä pystyy tarkistamaan ehdotusten oikeellisuuden sekä selaamaan käsin luokkarakennetta ja valitsemaan esineelle luokan, jonka ilmentymä se on.

2.2 Toimintaympäristö

Ohjelma suunnitellaan toimimaan sekä Linux että Windows-ympäristöissä. Ohjelma toteutetaan Java Server Pages(kts luku 8)- tekniikalla hyödyntäen JSP-merkkauksia (tageja)(kts luku 8) Jakarta Tomcat(kts. luku 8) -ympäristössä. XML-tiedostojen käsittelyyn tullaan käyttämään Sax(kts luku 8) ja Dom(kts luku 8) rajapintoja, RDF- tiedon käsittely toteutuu käyttäen Jena(kts. luku 8)- rajapintaluokkia. Ohjelmiston käyttöliittymä kuvataan XML:llä ja se suunnitellaan toimivaksi selaimella, kuten Netscape Navigator tai Microsoft Internet Explorer.

2.3 Käyttäjät

Ohjelman ensisijainen käyttäjäryhmä on museohenkilökunta, mutta ohjelma on syytä rakentaa tulevaisuutta ajatellen mahdollisimman yleiskäyttöiseksi. Metadataeditoria on tarkoitus käyttää Finnish museums online (FMO) -projektissa. Museohenkilökunta ei tiedä paljoakaan tietotekniikasta, mutta ymmärtää ontologian ja tavaroiden luokittelun tarkoituksen ja logiikan.

FMO-projektin ylläpidossa on tietotekniikkaa osaavia henkilöitä, jotka ymmärtävät XML:n sekä RDF(S):n toiminnan, ja osaavat museohenkilökunnan pyynnöstä tehdä muutoksia käytettävään ontologiaan.

2.4 Oletukset ja riippuvuudet

Koska Semantic Web-teknologiat ovat vielä uusi tietojenkäsittelyn osa-alue, on kaikkien ohjelmalta toivottujen vaatimusten toteuttamiseen kuluva aika vaikea arvioida.

Halutut ominaisuudet laitetaan prioriteettijärjestykseen ja tarvittaessa alhaisemman prioriteetin ominaisuuksia karsitaan.

3 Ohjelmistolle asetetut vaatimukset (M.A.)

Tämä luku kuvaa ohjelmistolle asetetut vaatimukset.

3.1 Järjestelmän tarjoamat palvelut

Järjestelmä tarjoaa käyttäjille seuraavat palvelut:

-- XML-dokumentin / kortin lukeminen, validointi halutun XML-skeeman mukaisesti ja muokkaus

Järjestelmä osaa lukea XML-muotoisen tiedoston, jonka sisältö on jaoteltu erilaisia kenttiä sisältäviksi ”korteiksi”. Järjestelmä osaa lukea XML-skeeman, jossa määritellään XML-kortin formaatti, ja tämän perusteella tarkistaa XML-kortin tietojen oikeellisuus. Järjestelmä osaa havaita kentissä olevat virheelliset arvot, joista käyttöliittymä antaa käyttäjälle visuaalista palautetta. Järjestelmän käyttöliittymä mahdollistaa XML-kortin tietojen muokkauksen.

-- Semantiikan liittäminen XML-korttiin annetun RDF(S)-muotoisen ontologian mukaisena

Järjestelmä osaa lukea ontologian RDF(S)-muotoisena tiedostona. Ontologia määrittelee käsitteistön ja niiden väliset suhteet, joiden perusteella luetut XML-kortit luokitellaan. Käyttäjän luokiteltua XML-kortin käyttöliittymää hyödyntäen järjestelmä osaa kirjoittaa RDF-muotoisen kuvauksen kortista. Järjestelmä ehdottaa automaattisesti luokituksia XML-kenttien perusteella.

-- Semanttisesti rikastetun XML-dokumentin tallentaminen haluttuun sijaintiin

Järjestelmä osaa kirjoittaa validoidun ja RDF-Skeeman mukaisesti luokitellun XML-kortin käsiteltävään XML-tiedostoon. RDF-muotoinen tieto luokituksesta kirjoitetaan XML-kortin viimeiseen kenttään. Käsiteltävän XML-tiedoston järjestelmä osaa tallettaa käyttäjän määrittelemään sijaintiin levyjärjestelmässä. Järjestelmä pystyy myös tallentamaan käsiteltävästä XML-tiedostosta valmiit kortit haluttuun sijaintiin levyjärjestelmässä.

-- Järjestelmä muistaa edellisen käsitellyn XML-kortin

Järjestelmä muistaa edellisen XML-kortin, jonka käyttäjä on luokitellut. Näin järjestelmän käyttäjä voi jatkaa työtään samasta kohdasta, mihin hän viimeksi lopetti työn. Järjestelmä tallentaa tilatiedon omaan sisäiseen tilatiedostoon.

-- Puutteellisten tietojen automaattinen lisäys

Järjestelmä osaa automaattisesti täyttää kortin tyhjiä kenttiä jos niille on määritelty oletusarvoja attribuutteina. Tämä tapahtuu siten, että järjestelmän havaitessa tyhjän kentän, se täyttää sen XML-Skeemassa sijaitsevalla, konfiguraatitiedostossa määritellyllä oletusarvo-attribuutin sisällöllä.

3.2 Lisäominaisuuksiin varautuminen

--Tekoäly

Ohjelmistoon tulee olla mahdollista jälkepäin lisätä älykkyyttä siten, että järjestelmän toiminta tehostuu sekä automaattisen objektien luokittelun, että puutteellisen XML-kortin kenttien automaattisen luokittelun osalta. Esimerkiksi järjestelmä osaisi automaattisesti päätellä uuden luokiteltavan XML-korttiin liittyvät ontologian luokitukset.

--Tietojen sijainti eri palvelimilla

Järjestelmä tulee olla laajennettavissa siten, että luettavat XML sekä RDF- tiedostot voivat sijaita verkkoyhteyden päässä eri palvelinkoneilla.

--Ontologiaan tehtävien muutosehdotusten kirjaus

Järjestelmään tulee olla mahdollista lisätä toiminto RDF(S)- muotoisen ontologian muutosehdotusten kirjaamiseksi, ja lähettämiseksi tietoliikenneyhteyden kautta toiselle taholle. Esimerkiksi museotyöntekijän luokitellessa esineitä, hän havaitsee että luokiteltavalle esineelle ei ole ontologiassa sopivaa luokkaa. Hän tekee uudesta luokasta ehdotuksen, jonka järjestelmä lähettää tietoliikenneyhteyden kautta ontologian laatimisesta vastaavalle henkilölle.

--Käyttöliittymä

Käyttöliittymä tulee olla toteutettu siten, että se on helposti muokattavissa/korvattavissa muihin kuin museotoimintoihin sopivaksi. Tällä tarkoitetaan sitä, että käyttöliittymän toteutus on irrallinen muista järjestelmän osista ja järjestelmän toimintoihin tarjotaan selkeä rajapinta, jolloin käyttöliittymä voidaan vaihtaa tarvitsematta puuttua muun järjestelmän toimintaa millään tavoin.

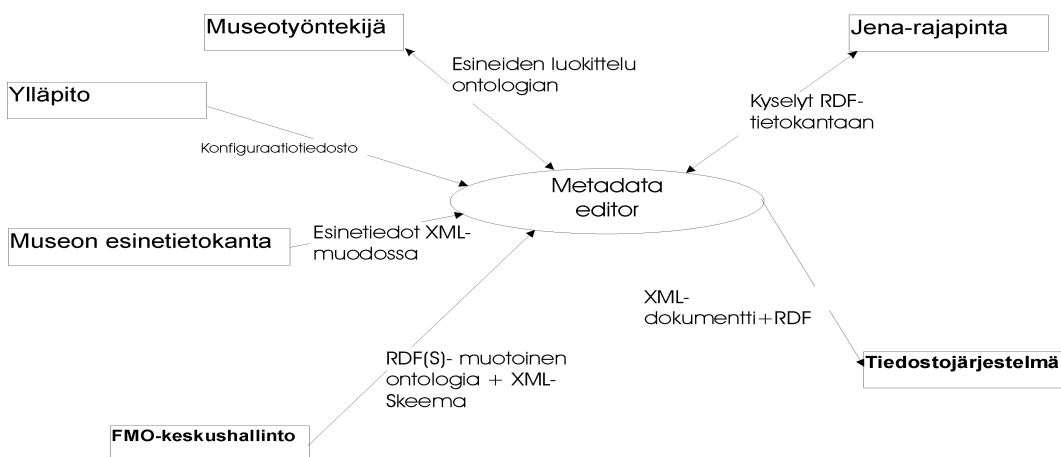
4 Toimintokuvaus (M.A.)

Tässä luvussa tarkastellaan rakennettavan järjestelmän sidosryhmiä, yleisarkkitehtuuria, toimintoja ja niiden toteutuksen tärkeysjärjestystä sekä määrittellään järjestelmän kannalta pakollisiksi sekä lisätoiminnoiksi luokiteltavat toiminnot.

4.1 Sidosryhmät

Järjestelmän sidosryhmät tuottavat järjestelmään tietoa ja syötteitä. Metadataeditori-järjestelmän sidosryhmiä ovat museotyöntekijät, FMO-projektin ylläpito, museon esinetietokanta, museo, Jena-rajapinta ja tiedostojärjestelmä.

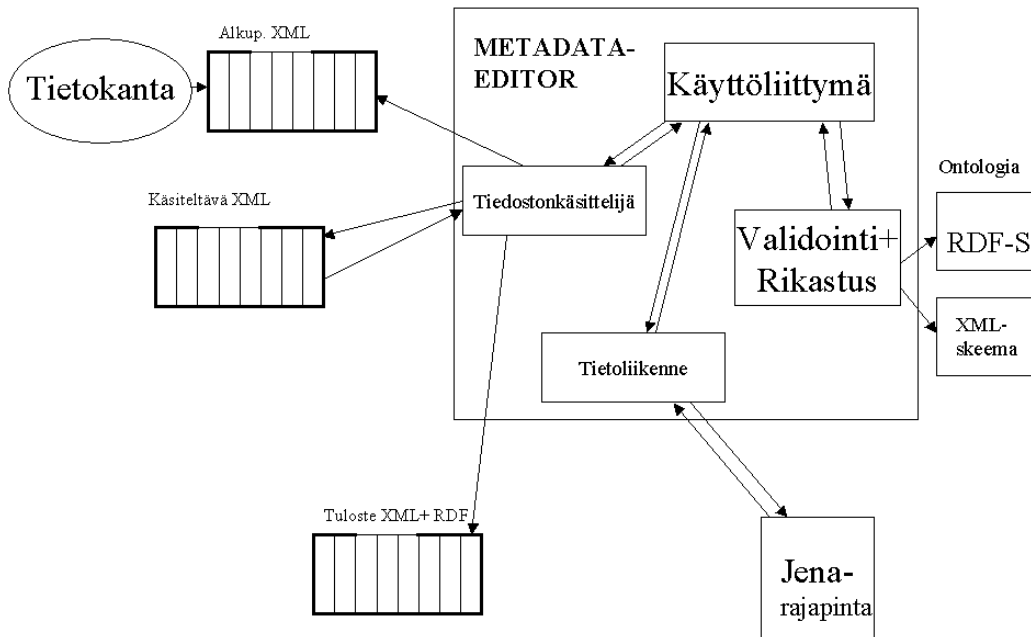
Kuva 4.1 havainnollistaa Metadataeditori-järjestelmän sidosryhmiä. Itse järjestelmä on kuvattu keskellä olevana ympyränä, jonka sisäiseen rakenteeseen ei tässä oteta kantaa. Järjestelmään tulevia tietovoita ovat museotyöntekijän käyttöliittymän kautta antamat komennot, ylläpidon laatimat konfiguraatiotiedostot, museon toimittama ontologia sekä museon esinetietokantaan pohjautuva XML-tiedosto. Järjestelmästä lähteviä tietovoita ovat puolestaan näytölle tulevat tulokset käyttäjän antamista komennoista, tiedostojärjestelmään tuotettava semanttisesti rikas XML-dokumentti, joka sisältää RDF-tiedon, sekä Jena-rajapintaan suoritettavat RDF-tietokannan kyselyt.



Kuva 4.1: Yhteyskaavio

4.2 Järjestelmän yleisarkkitehtuuri

Järjestelmä toimii siten, että ohjelma lukee tietokannasta tuotetun XML-kortteja sisältävän tiedoston, jonka se tallettaa omaa käyttöönsä varten erilliseen tiedostoon. Ohjelma lukee RDF(S)-muotoisen ontologian, ja käyttäjän valintojen mukaisesti tuottaa XML-dokumentin joka sisältää RDF-tietoa.



Kuva 4.2: Järjestelmän arkkitehtuuri

Järjestelmä toimii siten, että tietokannasta tuotettu alkuperäinen XML luetaan tiedostonkäsittelijä-osion toimesta sisään, ja siitä muodostetaan järjestelmän sisäinen kopio, johon tallennetaan korttikohtaiset tilatiedot ("käsittelyssä", "ei käsitelty", "valmis"). Tiedostonkäsittelijä antaa käyttöliittymälle XML-kortin, jonka käyttöliittymä antaa validoitavaksi/rikastettavaksi siitä vastaavalle järjestelmän osalle. Rikastuksesta/validoinnista vastaava järjestelmän osa lukee konfiguraatitiedoston määritelmien mukaisen RDF(S)-muotoisen ontologian, sekä XML-Skeeman, joiden mukaisesti validointi suoritetaan. Tämän lisäksi käyttäjä lisää korttiin RDF-Skeeman mukaisen luokittelun. Tähän toimenpiteeseen liittyy tietoliikenne-osion suorittamat kyselyt Jena-rajapinnan kautta saavutettavaan RDF-tietokantaan, josta tarkistetaan olemassaolevien ja uusien instanssien yhteensopivuus. Tiedostonkäsittelijä tallentaa käsitellyn kortin käsiteltävään XML-tiedostoon. Tiedostonkäsittelijä myös tallentaa haluttuun sijaintiin valmiin XML-tiedoston tai käsiteltävän XML-tiedoston valmiit kortit semanttisesti rikastettuina eli sisältäen RDF-tiedon.

4.3 Järjestelmän toiminnot

4.3.1 Konfiguraatitiedoston käsittely

Järjestelmä sisältää toiminnon konfiguraatitiedoston lukemista varten. Konfiguraatitiedostossa on määritelty järjestelmän toiminnan kannalta oleelliset asiat, järjestelmä ei toimi ilman konfiguraatitiedostoa. Tiedosto sisältää seuraavat tiedot: RDF(S)-muotoisen ontologiatiedoston nimi ja sijainti, XML-skeematiedoston nimi ja sijainti. XML-tiedostoon sisältyvien korttien alun ja lopun määrittelevän tagin nimi, kenttiin liittyvän ohje-attribuutin nimi, sekä kenttiin, että koko korttiin liittyvät otsikkoattribuuttien nimet, korttiin liittyvän kuvan sijainnin kertovan attribuutin nimi, sekä kortin että yksittäisen kentän julkaisukelpoisuus-attribuuttien nimet, ontologian älykkään täytön käyttö sekä oletusarvoattribuuttien automaattinen täyttö tyhjiin kenttiin päälle/pois-kytkentä.

Prioriteetti 1

1. RDF-skeeman sijainti
2. XML-skeeman sijainti
3. XML-kortin määrittelevät kentät (esim. <Kortti></Kortti>)
4. Otsikkoattribuutti yksittäiselle kentälle
5. Kuvan sijainnin määrittelevä attribuutti

Prioriteetti 2

1. Kentän julkaisukelpoisuus -attribuutti
2. Kortin julkaisukelpoisuus -attribuutti
3. Ontologian älykäs täyttö päälle/pois
4. Oletusarvoisten attribuuttien automaattinen täyttö päälle/pois
5. Tomcatin käynnistys/sammutus komento
6. Ohjeattribuutti (esim. <Kenttä ohje="Help"> missä *ohje* on ohjeen määrittelevän attribuutin nimi)
7. Otsikkoattribuutti koko kortille

```

; Metadata-editor ,
; konfiguraatiotiedosto (meedio.cfg)
;
RDF-SKEEMA=www.skeema.fi/rdfskeema.rdfs
XML-SKEEMA=www.skeema.fi/xmlskeema.xmls
TOMCAT=c:\tomcat\tomcat.bat
TOMCAT_OFF=c:\tomcat\tomcat_off.bat
XML-KORTTI=kortti
OHJEATTRIBUUTTI=ohje
KORTTI_OTSIKKOATTRIBUUTTI=kortti_otsikko
KENTTÄ_OTSIKKOATTRIBUUTTI=kentta_otsikko
KUVAN_SIJAINTIATTRIBUUTTI=kuvan_sijainti
;
KENTAN_JULKKELP_ATTRIBUUTTI=ke_julk_kelp
KORTIN_JULKKELP_ATTRIBUUTTI=ko_julk_kelp
ONTOLOGIAN_ALYKAS_TAYTTO=KYLLA
OLETUSARV_ATTRIB_TAYTTO=KYLLA
;

```

Esimerkki konfiguraatiotiedostosta

4.3.2 XML-validaattori

XML-Validaattori on järjestelmän sisäinen osa, joka lukee konfiguraatitiedostossa määritellyn XML-Skeeman, XML-kortin, tarkistaa korttien oikeellisuuden ja osaa toimittaa käyttöliittymälle tarkat tiedot XML-kortista ja sen tietojen oikeellisuudesta.

Prioriteetti 1

1. Osaa lukea XML-Skeeman
2. Osaa lukea XML-kortin
3. Osaa validoida XML-kortin halutun skeeman mukaisesti
4. Osaa antaa tarkat tiedot XML-kortista, ja sen tietojen oikeellisuudesta

4.3.3 RDF-käsittelijä

RDF-käsittelijä on järjestelmän sisäinen osa, joka osaa lukea RDF(S)-muotoisen ontologian sisäiseen tietorakenteeseen, joka näytetään käyttäjälle käyttöliittymän toimesta listatyypisenä rakenteena. RDF-käsittelijä osaa käyttäjän luokittelujen mukaisesti kirjoittaa XML-korttiin halutun ontologian mukaisen semanttisesti rikkaan RDF-tiedon. Järjestelmä osaa ehdottaa esineen nimen ja muiden XML-kenttien perusteella ontologiasta löytyviä samannimisiä luokkia automaattisesti. RDF-käsittelijä osaa tietoliikenne-osion avustuksella tarkistaa uusia instansseja tehtäessä Jena-rajapinnan kautta RDF-tietokannasta olemassa olevia instansseja ja verrata niitä keskenään.

Prioriteetti 1

1. Konfiguraatitiedostossa määritellyn RDF(S)-muotoisen ontologian lukeminen sisäiseen tietorakenteeseen
2. Jena-oliomallin mukaisen rakenteen muodostaminen RDF(S)-ontologiasta
3. Jena-oliomallin mukaisen rakenteen välittäminen käyttöliittymälle
4. Semanttisen kuvauksen tallennus XML-korttiin
5. Uusien instanssien vertaaminen olemassa oleviin instansseihin
6. Dokumenttien validiuden syntaktinen tarkastaminen

Prioriteetti 2

1. Semanttisen kuvauksen lukeminen XML-kortista
2. Semanttisen kuvauksen välittäminen käyttöliittymälle
3. Ontologian rajoitteiden huomiointi

4.3.4 Tiedostonkäsittelijä

Tiedostonkäsittelijä on järjestelmän osa, joka huolehtii XML-tiedostojen käsittelystä.

Prioriteetti 1

1. Osaa tehdä XML-tiedostosta kopion järjestelmän käyttöön
2. Osaa lisätä / muuttaa kortin tilatiedot (käsittelyssä / ei käsitelty / valmis)
3. Osaa lukea käsiteltävien XML-korttien ID-tiedot, otsikot ja tilatiedot
4. Osaa lukea kortin XML-tiedostosta
5. Osaa tallentaa kortin XML-tiedostoon
6. Osaa tallentaa tiedoston haluttuun paikkaan
7. Osaa tallentaa tiedostosta valmiit kortit haluttuun paikkaan
8. Osaa tuhota käsittelyssä olevan XML-tiedoston

4.3.5 Käyttöliittymä

Järjestelmän käyttöliittymä tarjoaa museotyöntekijöille tarkoitetun käyttäjäystävällisen ja selkeän mahdollisuuden XML-kortin tietojen validoimiseen halutun XML-skeeman mukaisesti ja kortin luokittelemiselle halutun ontologian mukaisesti. Käyttöliittymä näyttää kuvan esineestä, esineen tiedot selkeästi siten, että puutteelliset tiedot on maalattu punaisella värillä sekä ontologian selkeänä listarakenteena. Käyttöliittymä näyttää automaattisesti esineen nimen perusteella sopivat ontologian mukaiset luokkasuosituksen.

Prioriteetti 1

1. XML-kortin tietojen havainnollinen näyttäminen
2. XML-kortin tietojen muokkaus
3. XML-kortin virheellisten kenttien korostus
4. Ontologian esittäminen dynaamisena hierarkkisena listana
5. Semantiikan liittäminen XML-korttiin
6. Kortin tallennus
7. XML-tiedoston tallennus
8. Käsiteltävän XML-tiedoston valmiiden korttien tallennus
9. Kuvan näyttäminen käyttäjälle
10. Tiedoston sisältämien korttien näyttäminen havainnollisesti

Prioriteetti 2

1. XML-tiedostojen tiedostopolun määrittäminen (oletus määritellään konfiguraatitiedostossa)
2. Ontologian luokkien haku hakusanalla
3. Osaa näyttää kenttien oletusarvot
4. Osaa täyttää kenttiin niiden oletusarvot
5. Kortteihin liittyvien niiden käsittelyvaihetta kuvaavien väri- ja kirjainkoodien näyttäminen
6. Saman nimisten XML-kenttien ja RDF-luokkien kahteenkertaan näyttämisen estäminen
7. Osaa pyytää XML-validaattorilta kortin tiedot ja yhdistää ne ontologian luokkiin

4.3.6 Tietoliikenne-osio

Tämä järjestelmän osio osaa muodostaa yhteyden Jena-rajapinnan yli SQL-tietokantaan ja suorittaa siihen RDF-instanssihakuja. Järjestelmä osaa hakea RDF(S)-muotoisen ontologian verkkoyhteyden yli.

Prioriteetti 1

1. Kyselyn suorittaminen Jena-rajapinnan yli RDF-tietokantaan.
2. RDF-Skeematiedoston haku verkkoyhteyden yli

4.4 Järjestelmän toimintojen tärkeysjärjestys

Toiminnot jaetaan seuraaviin luokkiin:

1. *Pakolliset toiminnot.* Järjestelmän toiminnan kannalta kriittiset toiminnot. Ilman näitä toimintoja ei ole järjestelmää. Nämä toiminnot toteutetaan ehdottomasti.
2. *Lisätoiminnot.* Järjestelmän käytettävyyttä parantavat sekä merkittävää lisäarvoa tuovat toiminnot. Nämä toiminnot pyritään toteuttamaan projektin aikataulun puitteissa.
3. *Muut lisätoiminnot.* Vähemmän tärkeät toiminnot. Jatkokehitysehdotukset, joita suunnittelu ei sulje pois.

Taulukko 5.1 sisältää järjestelmään liittyvät toiminnot priorisoituina. (Taulukossa mainitut pakolliset toiminnot on kuvattu lähemmin luvussa 5.5. Lisätoiminnoiksi laskettavat toiminnot on puolestaan kuvattu tarkemmin luvussa 5.6.)

Prioriteetti	Toiminto
1	RDF-skeeman sijaintitiedon luku konfiguraatitiedostosta
1	Kortin alun ja lopun määrittelevät kentät (esim. <Kortti></Kortti>) (luku konfiguraatitiedostosta)
1	XML-skeeman sijaintitiedon luku konfiguraatitiedostosta
1	Otsikkoattribuutti yksittäiselle kentälle (luku konfiguraatitiedostosta)
1	Kuvan sijainti levyjärjestelmässä-attribuutti (luku konfiguraatitiedostosta)
1	Osaa lukea XML-Skeeman
1	Osaa lukea XML-kortin tiedostosta
1	Osaa validoida XML-kortin skeeman mukaiseksi
1	Osaa antaa tarkat tiedot XML-kortista, ja sen tietojen oikeellisuudesta
1	Konfiguraatitiedostossa määritellyn RDF(S)- muotoisen ontologian lukeminen

	sisäiseen tietorakenteeseen
1	Jena-oliomallin mukaisen rakenteen muodostaminen RDF(S)- ontologiasta
1	Jena-oliomallin mukaisen rakenteen välittäminen käyttöliittymälle
1	Osaa antaa käyttöliittymälle tiedot kohdan 5.5.16 mukaisista luokista
1	XML-tiedoston tallennus
1	Valmiin kortin tallennus
1	Kuvan näyttäminen käyttäjälle käyttöliittymässä
1	Tiedoston sisältämien korttien näyttäminen havainnollisesti
1	Osaa tehdä XML-tiedostosta kopion järjestelmän käyttöön
1	Osaa lisätä / muuttaa kortin tilatiedot (käsitellyssä / ei käsitelty / valmis)
1	Osaa lukea käsiteltävien XML-korttien ID-tiedot, otsikot ja tilatiedot
1	Osaa tallentaa kortin
1	Osaa tallentaa tiedoston haluttuun paikkaan
1	Semanttisen kuvauksen tallennus XML-korttiin
1	XML-kortin tietojen havainnollinen näyttäminen
1	XML-kortin tietojen muokkaus
1	XML-kortin virheellisten kenttien/attribuuttien korostus
1	Ontologian esittäminen dynaamisena hierarkkisena listana
1	Semantiikan liittäminen XML-korttiin
1	Osaa tallentaa valmiit kortit haluttuun paikkaan
1	Kortin tallennus
1	Uusien instanssien vertaaminen olemassa oleviin instansseihin
1	Kyselyn suorittaminen Jena-rajapinnan yli RDF-tietokantaan.
1	Dokumenttien validiuden syntaktinen tarkastaminen
1	RDF-Skeematiedoston haku verkkoyhteyden yli
1	Osaa tuhota käsiteltävän XML-tiedoston
2	XML-tiedostojen tiedostopolun määrittäminen (oletus määritellään konfiguraatitiedostossa)
2	Käyttöliittymä osaa pyytää XML-validaattorilta kortin tiedot ja yhdistää ne ontologian luokkiin
2	Semanttisen kuvauksen lukeminen XML-kortista
2	Kentän julkaisukelpoisuus- attribuutti (luku konfiguraatitiedostosta)
2	Kortin julkaisukelpoisuus-attribuutti (luku konfiguraatitiedostosta)
2	Ontologian älykäs täyttö päälle/pois (luku konfiguraatitiedostosta)
2	Oletusarvoisten attribuuttien automaattinen täyttö päälle/pois (luku konfiguraatitiedostosta)
2	Ontologian luokkien haku hakusanalla

2	Osaa näyttää kenttien oletusarvot
2	Osaa täyttää kenttiin niiden oletusarvot
2	Kortteihin liittyvien niiden käsittelyvaihetta kuvaavien värikoodien näyttäminen
2	Tomcatin käynnistys/sammutus komennon luku konfiguraatitiedostosta
2	Ohjeattribuutti (esim. <Kenttä ohje="Help"> (luku konfiguraatitiedostosta)
2	Semanttisen kuvauksen välittäminen käyttöliittymälle
2	Otsikkoattribuutti koko kortille (luku konfiguraatitiedostosta)
2	Saman nimisten XML-kenttien ja RDF-luokkien kahteen kertaan näyttämisen estäminen
2	Ontologian rajoitteiden huomiointi
Taulukko 5.1: Toimintojen prioriteetit	

4.5 Pakollisten toimintojen määrittely

Tässä luvussa esitellään taulukossa 5.1 esitellyt järjestelmän toiminnan kannalta pakolliset toiminnot.

-- RDF-skeeman sijaintitiedon luku konfiguraatitiedostosta

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatitiedostosta ontologian sisältävän RDF(S)-muotoisen tiedoston nimen ja hakemistopolun / URIn.

-- Kortin alun ja lopun määrittelevät kentät (esim. <Kortti></Kortti>) (luku konfiguraatitiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatitiedostosta tiedon XML-kortin alun ja lopun määrittelevistä kentistä. Konfiguraatitiedostoon tallennettujen kenttien nimien avulla kortit erotetaan XML-tiedostossa, joten ohjelma ei toimi ilman tätä tietoa.

-- XML-skeeman sijaintitiedon luku konfiguraatitiedostosta

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatitiedostosta XML-skeeman sisältävän tiedoston nimen ja hakemistopolun / URIn.

-- Otsikkoattribuutti yksittäiselle kentälle (luku konfiguraatitiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatiodostosta yksittäisen XML-kortin kentän otsikon määrittelevän attribuutin nimen. Kentistä etsitään konfiguraatiodostossa määritellyn nimistä otsikko-kenttää, joka näytetään käyttöliittymässä ennen kentän varsinaista sisältöä. Jos konfiguraatiodostossa määritellyn nimistä attribuuttia ei löydy, ei käyttöliittymässä näytetä mitään otsikkoa kenttien arvoille.

-- Kuvan sijainti levyjärjestelmässä -attribuutti (luku konfiguraatiodostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatiodostosta XML-korttiin liittyvän kuvan nimen ja hakemistopolun määrittävän attribuutin nimen. Kuvan URIn sisältämän attribuutin nimi määritellään konfiguraatiodostossa ja tämän nimistä attribuuttia haetaan kortin aloittavasta kentästä.

-- Osaa lukea XML-Skeeman

Järjestelmä osaa lukea XML-Skeeman halutun nimisestä tiedostosta ja tulkita sen sisällön.

-- Osaa lukea XML-kortin tiedostosta

Järjestelmä osaa lukea XML-tiedoston, ja erotella sen sisällön konfiguraatiodostossa määriteltävien kortin alku- ja loppu-tagien mukaisiin osiin. Järjestelmä osaa lukea näiden alku- ja loppu-tagien väliset yhden kortin kaikki tiedot sisäiseen tietorakenteeseen.

-- Osaa validoida XML-kortin skeeman mukaiseksi

Järjestelmä osaa verrata luetun XML-kortin syntaksin oikeellisuutta XML-tiedoston syntaksin määrittelevään XML-Skeemaan, sekä havaita ja järjestelmän käyttäjän avustuksella korjata puutteelliset ja/tai virheelliset tiedot.

-- Osaa antaa tarkat tiedot XML-kortista, ja sen tietojen oikeellisuudesta

Järjestelmän XML-tietojen lukemisesta ja validoinnista vastaava XML-validaattori-osio osaa antaa lukemastaan XML-kortista ulkopuoliselle kutsujalle tarkat tiedot jotka sisältävät jokaisen kentän sisältämät arvot, ja tiedot näiden arvojen syntaksin oikeellisuudesta konfiguraatiodostossa määritellyn XML-skeeman mukaisesti. Ulkopuolinen kutsuja on tämän järjestelmän tapauksessa käyttöliittymä.

-- Konfiguraatiodostossa määritellyn RDF(S)- muotoisen ontologian lukeminen sisäiseen tietorakenteeseen

Järjestelmä osaa lukea RDF-Skeeman halutun nimisestä tiedostosta sisäiseen tietorakenteeseen ja tulkita sen sisällön.

-- Jena-oliomallin mukaisen rakenteen muodostaminen RDF(S)- ontologiasta

Järjestelmä osaa muodostaa luetun RDF-Skeeman pohjalta Jena-oliomallin mukaisen rakenteen. Tällä tarkoitetaan sitä, että järjestelmä osaa lukea RDF-Skeeman sisäiseen tietorakenteeseen, jonka tarjoaa Jena-rajapinta.

-- Jena-oliomallin mukaisen rakenteen antaminen käyttöliittymälle

Järjestelmän RDF-tietojen käsittelystä vastaava osio osaa antaa lukemastaan RDF(S)- muotoisesta ontologiasta ulkopuoliselle kutsujalle Jena-oliomallin mukaisen rakenteen. Ulkopuolinen kutsuja on tämän järjestelmän tapauksessa käyttöliittymä.

-- Osaa antaa käyttöliittymälle tiedot ontologian luokista

Järjestelmän RDF-tietojen käsittelystä vastaava osio osaa antaa ulkopuolisille kutsujalle (joka tämän järjestelmän tapauksessa on käyttöliittymä) tiedon XML-kortin ontologiaan liittyvistä luokista.

-- XML-tiedoston tallennus

Järjestelmä osaa tallentaa käsittelyssä olevan XML-tiedoston haluttuun sijaintiin.

-- Valmiiden korttien tallennus

Järjestelmä osaa tallentaa käsiteltävän XML-tiedoston valmiiden XML-korttien tiedot omaan sisäiseen tiedostoonsa tai käyttäjän määrittelemään sijaintiin.

-- Kortin tallennus

Käyttöliittymä tarjoaa käyttäjälle toiminnon kortin tallentamiseksi.

-- Kuvan näyttäminen käyttäjälle käyttöliittymässä

Järjestelmän käyttöliittymä osaa näyttää XML-tiedostosta saatavan konfiguraatitiedostossa määritellyn kuvan sijainti-attribuutin perusteella haetun kuvatiedoston sisällön.

-- Tiedoston sisältämien korttien näyttäminen havainnollisesti

Järjestelmän käyttöliittymä osaa näyttää käyttäjälle XML-tiedoston sisältämien korttien tiedot havainnollisesti korostaen virheelliset kentät. Esimerkkitaipaus: kts Liite 1 (kuva käyttöliittymästä).

-- Osaa tehdä XML-tiedostosta kopion järjestelmän käyttöön

Järjestelmä osaa tehdä luettavasta XML-tiedostosta kopion järjestelmän omaa käyttöä varten, jolloin alkuperäistä tiedostoa ei muuteta.

-- Osaa lisätä / muuttaa kortin tilatiedot (käsitteilyssä / ei käsitelty / valmis)

Järjestelmässä XML-kortteihin liittyy tilatieto kortin käsittelyvaiheesta. Järjestelmä osaa lisätä ja muuttaa kortin tilatiedon käsiteltävän XML-tiedoston kortteihin. Vaihtoehtoisia tiloja ovat ”käsitteilyssä”, ”ei käsitelty” ja ”valmis”. Tilatiedoista ”käsitteilyssä”-tila on prioriteetin 2. toiminto.

-- Osaa lukea käsiteltävien XML-korttien ID-tiedot, otsikot ja tilatiedot

Järjestelmän XML-tiedostojen käsittelystä vastaava osio osaa lukea ja välittää ulkopuoliselle kutsujalle käsiteltävän XML-tiedoston korttien ID, otsikko ja tilatiedot. Ulkopuolisella kutsujalla tarkoitetaan tämän järjestelmän toista sisäistä osaa.

-- Osaa tallentaa tiedoston haluttuun paikkaan

Järjestelmä osaa tallentaa käsiteltävän XML-tiedoston halutun nimisenä haluttuun hakemistopolkuun.

-- Semanttisen kuvauksen tallennus XML-korttiin

Järjestelmä osaa tallentaa XML-kortin kenttään korttiin liittyvän RDF-muotoisen määritelmän kortin ontologian mukaisesta luokittelusta.

-- XML-kortin tietojen havainnollinen näyttäminen

Järjestelmän käyttöliittymä osaa näyttää havainnollisesti XML-kortin kaikkien kenttien arvot käyttäjälle.

-- XML-kortin tietojen muokkaus

Järjestelmän käyttöliittymä tarjoaa käyttäjäystävällisen tavan muokata XML-kortin kenttien ja attribuuttien arvoja.

-- XML-kortin virheellisten kenttien/attribuuttien korostus

Järjestelmän käyttöliittymä osaa näyttää käyttäjälle virheellisiä arvoja sisältävät XML-kortin kentät ja attribuutit korostetusti.

-- Ontologian esittäminen dynaamisena hierarkkisena listana

Järjestelmän käyttöliittymä osaa esittää RDF(S)- muotoisena luetun ontologian käyttäjälle dynaamisena hierarkkisena listana.

-- Käyttöliittymän toiminto semantiikan liittämiseksi XML-korttiin

Järjestelmän käyttöliittymä antaa käyttäjälle mahdollisuuden luokitella XML-kortti ontologian mukaisiin luokkiin.

-- Osaa tallentaa valmiit kortit haluttuun paikkaan

Tiedostonkäsittelijä osaa tallentaa kaikki käsitellyt XML-kortit halutun nimiseen tiedostoon, joka sijaitsee halutussa hakemistopolussa.

-- Osaa tallentaa kortin

Tiedostonkäsittelijä osaa tallentaa yksittäisen XML-kortin käsiteltävänä olevaan XML-tiedostoon.

-- Osaa tuhota käsittelyssä olevan XML-tiedoston

Tiedostonkäsittelijä osaa tuhota käsittelyssä olevan XML-tiedoston.

-- Uusien instanssien vertaaminen olemassa oleviin instansseihin

Kun käyttäjä lisää uuden instanssin, osaa järjestelmä verrata lisättävää instanssia tietoliikenne-osion kautta muodostettavan yhteyden avulla RDF-tietokannassa oleviin instansseihin. Tarvittaessa järjestelmä ehdottaa käyttäjälle samankaltaisia olemassa olevia instansseja, jotta yksikäsitteisyys jo luotujen instanssien kanssa varmistuu.

-- Kyselyn suorittaminen Jena-rajapinnan yli RDF-tietokantaan.

Tietoliikenne-osio muodostaa yhteyden RDF-tietokantaan Jena-rajapinnan avulla instansseihin liittyvien kyselyjen suorittamista varten. Tätä toimintoa käytetään uusien luotavien instanssien vertaamiseen jo olemassa oleviin.

-- Dokumenttien validiuden syntaktinen tarkastaminen

Järjestelmä osaa tarkistaa rikastetun XML-tiedoston syntaksin.

4.6 Lisätoimintojen määrittely

-- Kentän julkaisukelpoisuus -attribuutti (luku konfiguraatitiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatitiedostosta yksittäisen XML-kortin kentän julkaisukelpoisuuden määrittelevän attribuutin nimen.

-- Kortin julkaisukelpoisuus-attribuutti (luku konfiguraatitiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatitiedostosta XML-kortin julkaisukelpoisuuden määrittelevän attribuutin nimen.

-- XML-tiedostojen tiedostopolun määrittäminen (oletus määritellään konfiguraatitiedostossa)

Järjestelmän käyttöliittymä tarjoaa käyttäjälle mahdollisuuden määrittää luettavan XML-tiedoston nimen ja hakemistopolun. Kyseiset tiedot on oletuksena asetettu konfiguraatitiedoston määrittelyjen mukaisesti.

-- Ontologian älykäs täyttö päälle/pois (luku konfiguraatitiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatitiedostosta ontologian älykkään täytön pois/päälle-kytkevän tiedon.

-- Oletusarvoisten attribuuttien automaattinen täyttö päälle/pois (luku konfiguraatitiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatitiedostosta yksittäisen XML-kortin kentän automaattisen täytön pois/päälle-kytkevän tiedon.

-- Ontologian luokkien haku hakusanalla

Järjestelmän käyttöliittymä tarjoaa mahdollisuuden ontologian luokkien hakemiseen hakusanalla. Tämä ominaisuus parantaa käytettävyyttä etenkin tapauksessa, jossa ontologia on huomattavan laaja.

-- Osaa näyttää kenttien oletusarvot

Järjestelmän käyttöliittymä osaa näyttää XML-kortin kenttien oletusarvot.

-- Osaa täyttää kenttiin niiden oletusarvot

Järjestelmä osaa täyttää XML-kortin tyhjän kentän oletusarvolla, jonka määrää default-attribuutti.

-- Kortteihin liittyvien niiden käsittelyvaihetta kuvaavien värikoodien näyttäminen

Järjestelmän käyttöliittymä osaa näyttää korttien käsittelyvaiheen havainnollisesti värikoodien avulla käsiteltävän XML-tiedoston sisältämien korttien listassa.

-- Ontologian rajoitusten huomiointi

Järjestelmä osaa tunnistaa ontologiaan liittyvät rajoitukset ja estää rajoitusten vastaisen semantiikan lisäämisen. Esim. Helikopteri jonka materiaali on paperi olisi kelpaamaton tapauksessa, jossa on määritelty rajoite helikopterin materiaaleille.

-- Käyttöliittymä osaa pyytää XML-validaattorilta kortin tiedot ja yhdistää ne ontologian luokkiin

Järjestelmän RDF-tietojen käsittelystä vastaava osio osaa pyytää järjestelmän XML-validoinnista vastaavalta osiolta tiedot yhdestä XML-kortista, sekä yhdistää ne luettuun RDF(S)-muotoiseen ontologiaan. Tällä tarkoitetaan sitä, että järjestelmä osaa ehdottaa käyttäjälle XML-kortin kentän arvon kanssa samannimisiä ontologian luokkia.

-- Samanimisten XML-kenttien ja RDF-luokkien kahteen kertaan näyttämisen estäminen

Järjestelmä osaa tunnistaa XML-kortista ja RDF(S)-muotoisesta ontologiasta samannimiset kentät ja olla näyttämättä niitä kahteen kertaan.

-- Tomcatin käynnistys/sammutus komennon luku konfiguraatiotiedostosta

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatiotiedostosta Tomcat-palvelijan käynnistys-, sekä sammutuskomennon.

-- Ohjeattribuutti (esim. <Kenttä ohje="Help"> (luku konfiguraatiotiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatiotiedostosta yksittäisen XML-kortin kentän täyttöohjeen määrittelevän attribuutin nimen. Tämän tiedon avulla kortin kenttiin liitetty täyttöohje osataan näyttää käyttöliittymässä. Esimerkiksi kentän ”esineen ikä”- täyttöohje voisi olla ”Syötä tähän esineen ikä vuosissa”. Ohje näytetään käyttäjälle, jos tämä yrittää syöttää kenttään virheellisen arvon esim. ”tuhat vuotta”.

-- Otsikkoattribuutti koko kortille (luku konfiguraatiotiedostosta)

Järjestelmä osaa lukea tekstimuotoisesta konfiguraatiotiedostosta yhden XML-kortin otsikon määrittelevän attribuutin nimen. Tämän nimistä attribuuttia etsitään kortin alun määrittelevästä kentästä ja attribuutin arvo näytetään koko kortin otsikkona.

-- Semanttisen kuvauksen lukeminen XML-kortista

Järjestelmän RDF-käsittelijä osaa lukea talletetun XML-korttiin liittyvän kortin semanttisen kuvauksen. Toiminto mahdollistaa sen, että museon työntekijä voi tarkastella ja päivittää XML-korttiin aiemmin tehtyjä semanttisia luokituksia.

-- Semanttisen kuvauksen välittäminen käyttöliittymälle

RDF-käsittelijä osaa muuntaa semanttisen kuvauksen käyttöliittymäosan ymmärtämiksi tietorakenteiksi ja välittää nämä rakenteet käyttöliittymälle.

5 Kuvaus käyttöliittymästä (M.J. & J.K.)

Metadataeditorin käyttöliittymä on oleellinen osa ohjelmaa, koska ohjelman pitäisi olla mahdollisimman yleiskäyttöinen ja näin myös käyttöliittymältä vaaditaan selkeyttä, jotta eri alojen ihmiset, joiden ATK-taidot vaihtelevat suuresti, voisivat sitä käyttää. Toisaalta vaatimus käyttöliittymän taipumisesta erilaiseksi aina skeemojen ja ontologioiden mukaan pakottaa luultavasti ratkaisuihin, jotka eivät ole parhaita mahdollisia. Pyrimme käyttöliittymäsuunnittelussa tasapainoilemaan näiden toiveiden ja vaatimusten rajalla, jotta lopputulos olisi mahdollisimman hyvä.

5.1 Käyttäjryhmät

Koska metadataeditorimme tulee alustavasti olemaan nimenomaan museohenkilökunnan käytössä, otamme huomioon käyttäjryhmissä ainoastaan museohenkilökunnan sekä ylläpitäjät.

5.1.1 Museohenkilö

Museohenkilö käyttää metadataeditoria jotta saisi ihmisille enemmän tietoa välitettyä museon esineistä. Museohenkilö ei tunne ohjelman taustalla vaikuttavia tekniikoita kuten XML ja RDF, mutta oppii nopeasti ymmärtämään ontologian luokittelun, koska esineitä on museoissa luokiteltu ennenkin. Hän tietää paljon museoesineistä ja haluaa syöttää ohjelmaan oikeita tietoja. Mikäli museohenkilö sattuu olemaan kesätyöntekijä, joka on määrätty luokittelemaan esineitä, on toivottavaa, että hänelle annetaan tarkat kuvaukset esineistä sekä johdatus luokittelun toimintaan.

5.1.2 Ylläpitäjä

Metadataeditorin ylläpitäjä toimii FMO-projektissa, ja hänen oletetaan tuntevan XML- ja RDF/S-tekniikat ja hänen oletetaan myös osaavan tehdä muutoksia ontologiaan kun niitä tarvitaan sekä päivittämään konfiguraatiodostot ajan tasalle kun muutoksia tulee. Koska Tietojenkäsittelytieteen laitos on mukana FMO-projektissa, oletamme ylläpitäjän olevan joku täältä ja näin voidaan olettaa, että hän tuntee hyvin kaikki seikat, jotka aihepiiriin kuuluvat.

5.2 Käyttötapaukset

-- Käyttötapaus I

Käyttäjän tavoite: Museotyöntekijä Matti on ollut työssä museossaan jo parikymmentä vuotta, ja tuntee esineistön paremmin kuin kukaan muu. Osaamisensa vuoksi Matti on valittu museonsa FMO-vastaavaksi, ja on juuri saanut ohjelman asennettua museon koneelle. Matti haluaa tutustua ohjelmaan, ja haluaa heti lähteä lajittelemaan museonsa kokoelmia.

Statustietoa:

- Matti ei tunne tietotekniikkaa paljoakaan, mutta on kiinnostunut oppimaan
- Metadataeditori toimii museon koneella, mutta Matti ei ole koskaan käyttänyt sitä
- Ontologia on tehty vastaamaan suunnilleen perinteisiä museo-luokitteluja

-- Käyttötapaus II

Käyttäjän tavoite: Matti on jo opetellut käyttämään metadataeditoria, kun häneltä pyydetään kansainväliseen näyttelyyn kaikkia heidän museonsa aseita. Matti haluaa löytää kaikki aseet heidän tiedoistaan ja haluaa liittää niihin luokittelut, ja tarkastaa että niiden tiedot ovat muutenkin ajan tasalla.

Statustietoa:

- Matti osaa käyttää ohjelmaa
- Matilla on paperilistaus kaikista aseista ja niiden numeroista
- Kaikista aseista ei tiedetä onko niitä käytetty myös muihin tarkoituksiin (työkaluina, uskonnollisiin menoihin...)
- Useiden aseiden tiedot ovat puutteellisia (ei tiedetä ikää, joistain on kadonnut tieto löytäjästä)

-- Käyttötapaus III

Käyttäjän tavoite: Mirkku on päässyt kesätöihin museoon jo kolmatta kertaa. Hän on aikaisimpina vuosina syötellyt tietoja museon tietokantaan, mutta nyt museossa on uusi ohjelma, jota käyttäen Mirkun pitäisi tarkistaa tiedot ja lisätä niihin luokituksia.

Statustietoja:

- Mirkku ei ole ennen käyttänyt ohjelmaa
- Mirkku ei ole aiemmin luokitellut museoesineitä
- Mirkku on käyttänyt tietokonetta kotona surffailuun ja sähköpostin lukemiseen

-- Käyttötapaus IV

Käyttäjän tavoite: Antti on Keltaiset sivut-projektiryhmän jäsen, ja haluaa tutkia miten heidän ryhmänsä voisi käyttää metadataeditoria hyödykseen projektissaan.

Statustietoja:

- Antti tuntee hyvin RDF(S)-tekniikan sekä muutakin metadataeditorissa käytettyä tekniikkaa
- Keltaiset sivut-projekti tarvitsisi työkalun, jolla ilmoittajat voivat laittaa keltaisille sivuille ilmoituksia RDF-muodossa käyttäjäystävällisesti

Anttia kiinnostaa erityisesti, miten museokäyttöön tarkoitettu editori toimii täysin uudella ja erityyppisellä ontologialla ja XML-skeemalla

5.3 Toteutettavat osat

Käyttöliittymä koostuu alustavasti neljästä osasta:

-- Muokattavan XML-kortin valinta

Käyttöliittymässä sijaitsevan kentän avulla käyttäjä voi hakea muokattavan tiedoston ja tiedostosta halutun XML-kortin. Tiedostot ovat näkyvissä tiedostot otsikon alapuolella kentän vasemmassa reunassa. Kun käyttäjä on valinnut tiedoston, ilmestyy kentän vasempaan reunaan lista tiedoston sisältämistä XML-korteista. Käyttäjä voi listasta valita haluamansa kortin muokattavaksi. Sekä tiedostot, että XML-kortin on värikoodattu siten, että käyttäjä näkee mitkä tiedostot/kortit on jo käsitelty.

-- XML-kortin muokkaus rdf-skeeman mukaiseksi

Oleellinen osa käyttöliittymää tulee olemaan hierarkkinen lista, joka muodostetaan ontologian perusteella. Sitä pitää pystyä selaamaan ja siitä pitää voida merkitä luokka, jonka ilmentymä esine on. Valintojen perusteella lisätään korttiin kentät, joista voi muuttaa myös attribuuttien arvoja, sekä uusia luokkia mikäli attribuutin arvona voi olla luokka. RDF- ja XML-tiedot pitää myös voida tallentaa. Tähän osaan saatetaan myös lisätä jonkinlainen haku, jolla käyttäjän ennalta tietämä luokka voidaan hakea ja kenties merkata suoraan. Järjestelmään kuuluu myös XML-kenttien arvoista luokkia päättelevä ohjelman osa, joka helpottaa luokittelutyötä.

--Museoesineen kuva

Kuvan näyttäminen on yksi käyttöliittymän tehtävistä. Kuvan ruudulle piirtämisestä vastaa selain.

-- XML-kortin muokkaus XML-skeeman määrittelemään muotoon

Oleellisena käyttöliittymän osana on ikkuna, joka näyttää XML-kortin sisältämät tiedot. Näytettävät kentät määräytyvät XML-skeeman mukaan. Korttia ladattaessa validaattori antaa tiedot virheellisistä tai puuttuvista tiedoista ja taulukko, johon tiedot tulevat, ilmoittaa käyttäjälle vajaista/virheellisistä tiedoista vaihtamalla solun taustaväriä. Mikäli XML-skeemaan on lisätty ohje-attribuutteja, tulostaa käyttöliittymä myös ohjeen kentän oikeasta täyttämisestä. Korttiruutuun ilmestyvät myös RDF-tiedot luokkavalintojen perusteella, ja niiden attribuutteja voi muokata tästä ruudusta.

5.4 Toteutustekniikat

Käyttöliittymä toteutetaan JSP-sivuina käyttäen JSP/Tag libraries-tekniikkaa. JSP:llä käyttöliittymä generoidaan HTML(kts luku 8) muotoon ja CSS(kts. luku 8)-tekniikkaa tullaan myös ehkä käyttämään. Mahdollisesti käytetään myös JavaScripteja(kts. luku 8), appletteja(kts. luku 8) tai servlettejä(kts. luku 8). Käyttöliittymä generoidaan XML- ja RDF-skeemojen pohjalta.

CSS-tekniikkaa hyödynnetään käyttöliittymän ulkoasun laadinnassa. Mahdollisia JavaScripteja, appletteja tai servlettejä hyödynnetään käyttöliittymässä, näiden tekniikoiden avulla on tarkoitus parantaa järjestelmän käytettävyyttä.

5.5 Erityisvaatimukset

Jotta käyttöliittymästä saataisiin selkeä, toivotaan ohjelmalle syötettävältä XML-skeemalta tiettyjä asioita. Nämä asiat vaikuttavat ratkaisevasti ohjelman käytettävyyteen.

- kentillä joiden tieto halutaan näkyviin tulee olla ”otsikko”-attribuutti, joka voidaan sitten käyttöliittymässä tulostaa ennen varsinaista tietoa
- jotta attribuutit voisi järkevästi kuvata käyttöliittymään, tulisi niiden nimien olla kuvaavia

- mikäli käyttäjät haluavat ohjeita ei-validien XML-kenttien korjaamiseen, on jokaiseen elementtiin XML-skeemassa lisättävä ohje-attribuutti

6 Muut ominaisuudet (J.K.)

Kappaleessa kerrotaan ohjelmistossa esiintyvistä muista ominaisuuksista.

6.1 Ulkoiset liittymät

-- Laitteistoliittymät

Ohjelmiston käyttämiseksi tietokoneessa tulisi olla tietoliikenneyhteys, hiiri, näyttö ja näppäimistö.

-- Laitteistorajoitukset

Ohjelma ei toimi ilman tietoliikenneyhteyttä. Tietoliikenneyhteys on välttämätön, jotta päästään käsiksi RDF-tietokantaan Jena-rajapinnan kautta.

6.2 Ylläpidettävyys

Ohjelmiston mukana tulevan dokumentaation avulla ohjelmistoa voidaan jatkokehittää. Dokumentaatio sisältää käyttöohjeen, testaussuunnitelman sekä määrittely-, suunnittelu-, toteutus-, ja testausdokumentit. Tuotettavat dokumentit on lueteltu ryhmän projektisuunnitelmassa. Dokumentointikielenä on Suomi ja dokumentit laaditaan Microsoft Word-97 ohjelmalla. Dokumentit julkaistaan myös PDF-muodossa.

Meedio-projektiryhmä ei ole vastuussa ohjelmiston ylläpidosta.

6.3 Siirrettävyys, yhteensopivuus

Ohjelma suunnitellaan ensisijaisesti toimivaksi Windows ja Linux ympäristöissä Netscape Navigator ja Internet Explorer selaimilla. Ohjelmisto lienee Java-toteutuksesta seuraavan yleiskäyttöisyyden vuoksi helposti siirrettävissä muihinkin käyttöjärjestelmiin, joihin on saatavilla Java-1.3.1-ympäristö sekä Tomcat palvelinohjelmisto. Siirrettävyyttä ei kuitenkaan taata.

Järjestelmässä on varauduttu siirrettävyyteen muille kielille (esim. englannille) siten, että haluttaessa ohjelman toimivan vieraalla kielellä laaditaan halutulla kielellä XML-skeema.

7 Testaus (J.K.)

Tässä kappaleessa kuvataan järjestelmän testauksessa käytettävät menetelmät.

7.1 Järjestelmätestaus

Järjestelmätestauksessa varmistetaan ohjelmassa laadittujen moduulien toiminta yhdessä ja erikseen. Testauksen tarkoituksena on saada ohjelmisto toimimaan asiakkaan haluamalla tavalla. Testaus suunnitellaan siten, että käyttöliittymän testaus ja ohjelmiston teknisten toimintojen testaus voidaan suorittaa erikseen. Kun molemmat osa-alueet on testattu ne yhdistetään, minkä jälkeen testataan ohjelmiston toimintaa kokonaisuutena. Testauksesta laaditaan testausdokumentti.

-- Testaajat

Kaikki ryhmän jäsenet osallistuvat testaukseen. Jokainen testaa tekemänsä moduulin itse välittömästi sen valmistuttua. Moduulien yhdistämisen yhteydessä tapahtuvasta integrointi-testauksesta vastaa Ari Inkovaara. Käyttöliittymän testauksesta vastaavat Justus Karekallas ja Miikka Junnila.

-- Menetelmät

Moduulitestauksessa painotetaan rakenteellista (white-box) testausta. Käytännössä tämä tarkoittaa sitä, että moduulille laaditaan pääohjelma, jonka avulla suoritetaan sekä peruspolkustaus, ehtotestaus että silmukkatestaus. Moduulin valmistuttua järjestetään FTR-kokous, jossa päätetään sen kelvollisuus.

Kun tietyn kokonaisuuden muodostavat moduulit on saatu valmiiksi, suoritetaan toiminnallinen (black-box) testaus. Tarkoituksena on selvittää moduulien keskinäisen toiminnan toteutumista suunnitellulla tavalla. Myös tässä yhteydessä järjestetään FTR-kokouksia, joilla varmistetaan testattujen kokonaisuuksien laatu.

Integrointitestaus aloitetaan kaikkien moduulien valmistuttua. Integrointitestauksen tarkoituksena on yhdistää moduulit yhdeksi toimivaksi ohjelmisto-kokonaisuudeksi. Käytettävä menetelmä on bottom-up testaus, jota tukee mm. edellä mainittu toiminnallinen testaus. Bottom-up testauksella ohjelmistoa lähdetään rakentamaan alhaalta ylös moduuleja yhdistämällä, kunnes saadaan koko ohjelmisto kokoon.

7.2 Muut testausvaiheet

Integrointitestauksen valmistuttua suoritetaan validointitestaus, jonka tarkoituksen on selvittää, että ohjelmisto toimii asiakkaan esittämien vaatimusten mukaisesti. Testausmetodina on tarkoitus käyttää toimintotestausta. Tätä varten laaditaan testausmateriaali siten, että testaus kattaa koko ohjelmiston toimialueen. Testausmateriaalina käytetään aitoa kansallismuseon materiaalia.

8 Viittaukset lähteisiin

Tässä luvussa kuvataan dokumentissa esiintyvät viittaukset eri tekniikoihin.

- XML (Extensible Markup Language), lisätietoa: <http://www.w3.org/XML/>

- XML-Skeema, lisätietoa: <http://www.w3.org/XML/Schema>
- RDF (Resource Description Framework), lisätietoa: <http://www.w3.org/RDF/>
- Java Server Pages, lisätietoa: <http://java.sun.com/products/jsp/index.html>
- JSP+tag libraries, lisätietoa: <http://java.sun.com/products/jsp/taglibraries.html>
- Jakarta Tomcat, lisätietoa: <http://jakarta.apache.org/tomcat/index.html>
- Sax (Simple API for XML), lisätietoa: <http://www.saxproject.org/>
- Dom (Document object model), lisätietoa: <http://www.w3.org/DOM/>
- Jena, lisätietoa: <http://www.hpl.hp.com/semweb/jena-top.html>
- HTML (Hypertext Markup Language), lisätietoja: <http://www.w3.org/MarkUp/>
- CSS (Cascading Style Sheets), lisätietoja: <http://www.w3.org/Style/CSS/>
- Java Applet, lisätietoja: <http://java.sun.com/>
- Java Servlet, lisätietoja: <http://java.sun.com/>