Toteutusdokumentti

WWW-museo

Tekijät:

Kim Josefsson (KJ)
Samppa Saarela (SS)
Eetu Mäkelä (EM)
Taneli Rantala (TR)
Robert Holmberg (RH)
Pasi Lehtimäki (PL)

Helsinki 24. toukokuuta 2002

Versio 0.8

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

1 Johdanto

Tämä on toteuttamamme järjestelmän toteutusdokumentti. Projektissa toteutettiin kaksi prototyyppiä. Ensimmäinen prototyyppi hylättiin, koska siinä oli monia heikkouksia, joita käsitellään tarkemmin luvussa 7. Toinen prototyyppi on parempi ja olemme dokumentoineet sen tarkasti tässä dokumentissa.

Luvussa 2 esitetään yleiskuvaus järjestelmästä. Luvussa käydään läpi mm. kaikki moduulit ja pakkaukset. Luku antaa hyvän pintapuolisen kuvan järjestelmästä. Luvussa 3 esitellään yksityiskohtaisesti metatietomoduuli. Moduulin luokat käydään läpi ja luokista esitetään luokkaaviot. Luvussa 4 esitellään suodatinmoduuli. Luvussa 5 esitellään profiilimoduuli. Luvussa 6 esitellään käyttöliittymämoduuli. Luvussa 7 käsitellään projektin aikana toteutettuja kahta prototyyppiä. Luvussa perustellaan, miksi toinen prototyyppi päätettiin tehdä ja mitkä ovat kummankin prototyypin hyvät ja huonot puolet. Luvussa 8 käsitellään järjestelmän testausta. Luvussa kerrotaan, miten ja kuinka kattavasti testasimme ohjelmiston toimintaa. Jokaista moduulia käsitellään erikseen. Luvussa 9 arvioidaan toteutuksen onnistumista, tuodaan esiin toteutuksen ongelmia ja esitetään parannusehsotuksia ja mahdollisia lisäominaisuuksia.

1.1 Versiohistoria

Versio	Päivämäärä	Tehdyt muutokset
0.8	6.5.2002	Alustava versio. Sisältää perusrakenteen.
0.9	24.5.2002	Lähes valmis versio.

2 Järjestelmän yleiskuvaus (SS)

Tässä luvussa käsitellään W_3M-järjestelmän komponentit yleisellä tasolla. Luvussa 2.1 käsitellään järjestelmän ohjelmalliset komponentit ja luvussa 2.1.6 näiden jakautuminen Javan pakkauksiin. Luvuissa 2.2, 2.3 ja 2.4 käsitellään järjestelmän tietosisältö (metadata).

2.0.1 Apuohjelmat (SS)

Jena (+Xerces) Protege Apache Tomcat J2EE (vähintään JSP 1.1) ANT

2.1 Osajärjestelmät (SS)

Tässä luvussa kuvataan W_3M-järjestelmän Java-komponentit yleisellä tasolla sekä näiden integraatio.

Ohjelmisto on jaettu neljään moduuliin, jotka edelleen jakautuvat alimoduleiksi. Järjestelmän moduulit ovat Metatieto-, Suodatin-, Profiili- ja Käyttöliittymämoduuli.

Näiden ohjelmamoduulien lisäksi järjestelmään kuuluu yleiskäyttöisiä Java Beans-nimeämiskäytönnön mukaisia apuluokkia, joita käytetään käyttöliitymän JSP-osan toteutuksessa. Java-pakkaukset on kuvattu luvussa 2.1.6.

2.1.1 Metatietomoduuli (SS)

Metatietomoduuli on koko järjestelmän ydin ja tarjoaa rajapinnan ontologian käsittelyyn verkkona, joka koostuu pääpiirteittäin luokkamäärittelyistä, ominaisuuksista, instansseista ja kuvista.

2.1.2 Suodatinmoduuli (SS)

Suodatimoduuli lisää Metatietomoduulin tietosisältöön utiliteettiarvot. Utiliteettiarvot perustuvat ontologiaverkon solmujen läheisyyteen, jota on mahdollista mitata esimerkkiksi suosituimmuuteen perustuen. Suodatinmoduuli tarjoaa myös näihin utiliteettiarvoihin pohjautuvan ontologiaverkon solmujen järjestämispalvelun.

2.1.3 Profiilimoduuli (SS)

Profiilimoduulin tietosisältönä on käyttäjäkohtainen profiili ja tilatiedot, joita suodatin käyttää hyväksi utiliteetien laskemisessa ja käyttöliittymä HiBrowsen toteutuksessa.

2.1.4 Käyttöliittymämoduuli (SS)

Käyttöliittymämoduuli tarjoaa kaksi erilaista liittymää ontologiapohjaisesti annotoitujen kuvien hakemiseen ja katseluun sekä ylläpitää profiilia.

2.1.5 Moduulijaon perustelut (SS)

2.1.6 Pakkaukset

Ohjelmakoodin eli Java-luokkien sijoittelu pakkauksiin (package) ilmentää pääosin yllä kuvattua moduulijakoa.

2.2 Sisällölliset ontologiat

Tässä tapauksessa Promootio-ontologia, mutta tämä on vaihdettavissa.

2.3 Kuvakortit (SS)

2.4 Sisäiset ontologiat (SS)

Varsinaisen näyttelyn aihepiiriä kuvaavan ontologian lisäksi järjestelmä käyttää muutamia vain järjestelmän sisäiseen käyttöön tarkoitettuja ontologioita. Näiden ontologioiden avulla määritellään fasettien juuret, kuvatiedostojen tekniset tiedot ja ontologiaelementtien painot. Nämä ontologiaskeemat ovat muuttumattomia.

2.4.1 Fasetti (RH)

Fasettiontologia on tapa lukea ja tallentaa perusontologian fasettijuuriksi valitut solmut Jenan avulla.

2.4.2 Kuvatiedosto (RH)

Kuvatiedosto-ontologia on rakenteeltaan hyvin yksinkertainen. Siihen tallennetaan kuvien kuvatiedostojen tietoja. Kuvatiedostoja voi jokaisella kuvalla olla useampi, ja ontologiaan tallennetaan myös kuvatiedostojen teknistä tietoa. Kuvatiedoston teknisinä ominaisuuksina ovat tiedostonimi, perus-URL, korkeus, leveys, väritieto ja tiedostomuoto.

2.4.3 Paino (RH)

Paino-ontologia on myös rakenteeltaan yksinkertainen ja sitä käytetään järjestelmän ontologiaresurssien painojen lukemiseen ja tallentamiseen.

2.4.4 Arvio (SS)

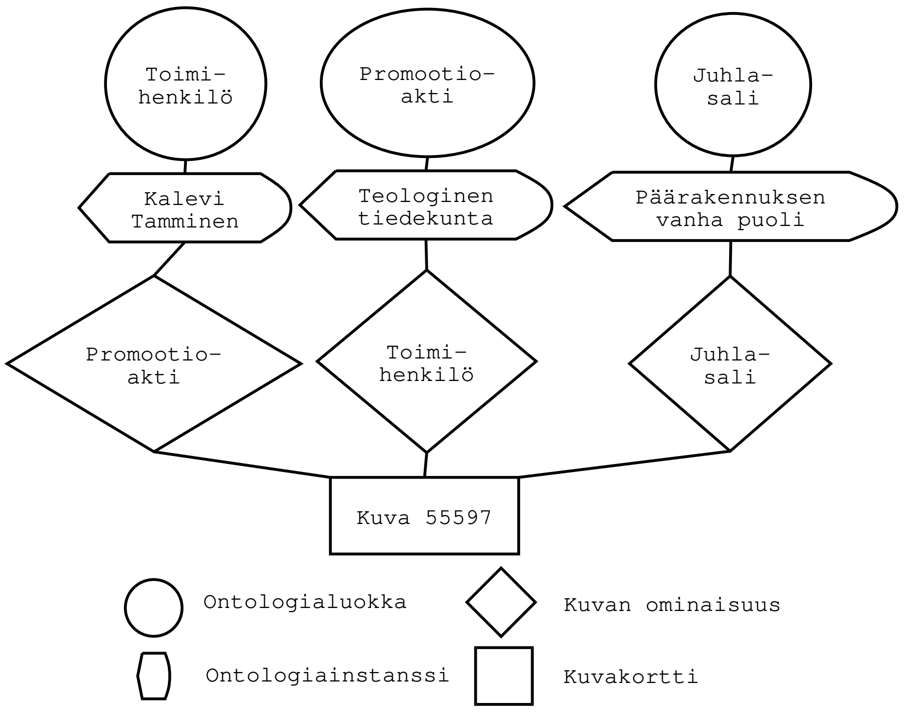
Kiinnitettyjä sisäisiä fasetteja on liikaa ja niitä on mahdollista käsitellä yleisemmällä tasolla.

Kuvatiedosto Tämän hetkinen toteutus kiinnittää kuvattavan asian kuvaksi ja jos myöhemmin haluttaisiin lisätä näyttelyyn esimerkiksi videoita, audiota tai dokumentteja, täytyy jokaiselle näistä tehdä oma Item-tyyppinen (Java) käsittelijänsä.

Dynaamisemman järjestelmästä saa antamalla mahdollisuuden määritellä erilaisia kuvattavia asioita (kohteita) ontologian avulla. Varsinaiset kohteet määriteltäisiin tällöin Item-luokan (RDFS) aliluokkina. Jos kohteena on kuva, varsinaiset kuvat olisivat Item-luokan perivan Image-luokan ilmentymiä. Näin ollen Kuvatiedosto-ontologia ei olisi enää järjestelmän sisäinen ontologia ja periaatteessa korvattavissa.

3 Metatietomoduuli (RH, KJ)

Luvussa 3.1 on kuvattu metatietomoduuli yleisellä tasolla. Luvussa 3.2 esitellään järjestemän käyttämät ontologiat. Luvussa 8.2 esitellään metatietomoduulin muille moduuleille tarjoamat palvelut. Luvussa 3.4 kuvataan metatietomoduulin ulkoisten palveluiden toiminta. Luvussa 3.5 kuvataan metatietomoduulin luokat ja metodit, joskaan ei pseudokooditasolla.



Kuva 1: Kuvakortti joka liittyy kolmeen ontologialuokan, ominaisuuksiensa ja ontologian ilmentymien kautta.

3.1 Yleiskuvaus

Metatietomoduuli luo, ylläpitää ja tallentaa oliomallin käytetyistä ontologioista, ontologialuokkien ilmentymistä ja kuvakorteista. Moduuli tarjoaa korkeamman tason palveluja ohjelmiston muille osille. Ontologioiden hallintajärjestelmänä käytetään Jena-kirjastoa. Kuvakortit, ontologiat ja ontologoiden luokkien ilmentymät ovat palvelimen levyllä RDF-muodossa. Itse kuvatiedostot ovat erillisessä hakemistossa palvelimen levyllä. Näitten välinen suhde on visualisoitu kuvassa 1.

3.2 Ontologiat

Järjestelmässä on neljä ontologiaa: perusontologia, fasettiontologa, kuvatiedosto-ontologia ja paino-ontologia. Tässä toteutuksessa käytettiin perusontologiana Avril Styrmanin tekemää promootio-ontologiaa. Se on rakenteeltaan metsä, eli yksinkertaisempi kuin ontologian perusrakenne, suunnattu verkko. Tämän vuoksi saattaa toteutuksessa olla ongelmia jotka eivät ilmene käytetyllä perusontologialla. Muut ontologiat kuvattiin luvussa 2.3. Jokaisella perusontologian instanssilla on syytä olla 'Nimi'-atribuutti, joka sisältää instanssin kuvauksen, tai nimenä näkyy instanssin uri.

3.3 Moduulin ulkoinen rajapinta

Tässä luvussa kerrotaan mihin metatietomoduulia käytetään ja miten. Metatietomoduulin luokkakaavion löytyy kuvassa 2 ja ontologiaresurssien käsittelyyn tehtyjen työkalujen luokkakaavio kuvassa 3.

3.3.1 Mallin luominen ja alustaminen

Luodaan MuseumModel-olio. Konstruktorin ensimmäisenä parametreina annetaan java.net.URL-olio, jonka sisältämän urlin alta löytyy kuvat sisältävä pictures/kuvatyyppi -hakemistorakenne. Kuvatyyppi on kokonaisluku, eli hakemistot ovat `pictures/0`, `pictures/1`, jne. Kuvatiedostojen oletetaan löytyvän täältä mikäli niille ei ole annotoitu omaa urlia. Konstruktorin toisena parametreina annetaan merkkijono, joka sisältää paikallisen hakupolun hakemistoon jonka alta löytyy `rdf/` ja `rdf/ontology`- hakemistot. `rdf/` -hakemistosta löytyy kuvien annotointidata `img.rdf`- tiedostossa. `rdf/ontology` -hakemistossa löytyvät ontologiat rdfs-tiedostoissa ja niitten instanssit rdf-tiedostoissa.

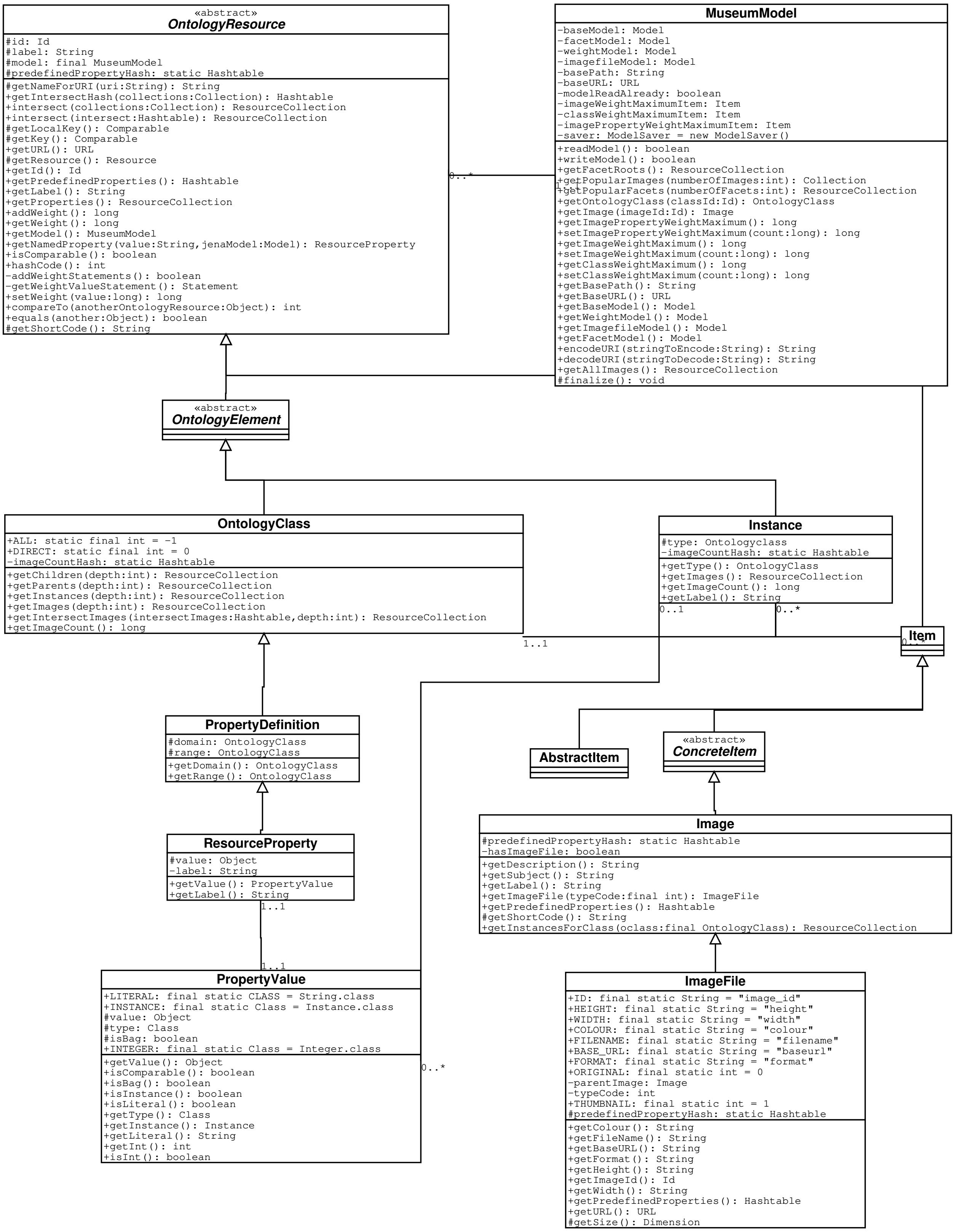
MuseumModel-oliolle kutsutaan readModel()-metodia, joka lukee tiedostot levyltä ja Jenan avulla rakentaa näistä oliomallin. Mallia ei tarvitse kirjoittaa erikseen levylle, koska se kirjotetaan automaattisesti tunnin väliajoin ja lisäksi kun MuseumModel-olio tuhotaan.

Esimerkki

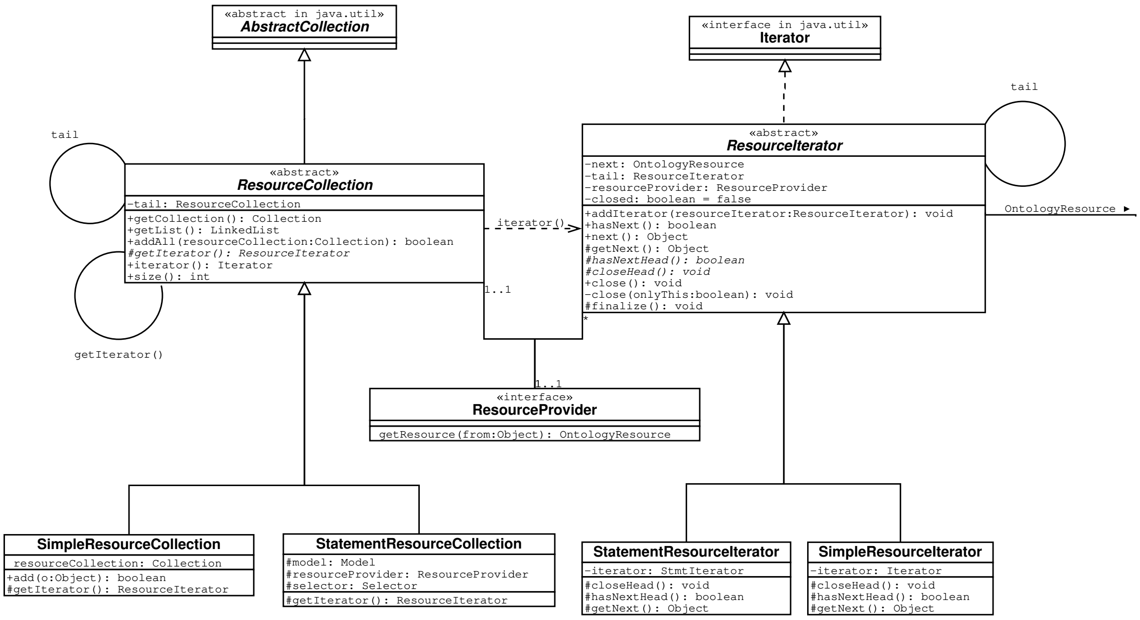
```
MuseumModel model = new MuseumModel(new URL(
    "http://localhost/"), "data/");
model.readModel();
```

3.3.2 HiBrowsen fasettipuiden hakeminen

Luodusta MuseumModel-oliosta voidaan hakea fasetitpuut hakemalla ensin fasettijuuret MuseumModelin getFacetRoots()-metodilla. Fasettijuuret ovat OntologyClass-



Kuva 2: Metatiedon luokkakaavio



Kuva 3: Ontologiaresurssit

olioita. Tämän jälkeen puut voidaan rakentaa rekursiivisesti kutumalla OntologyClass-olioiden getChildren()-metodia.

Esimerkki

```
Iterator facetIterator =
    model.getFacetRoots().iterator();
while (facetIterator.hasNext()) {
    OntologyClass oclass =
        (OntologyClass)facetIterator.next();
    //..hae tälle juurelle alipuu.
}
```

3.3.3 HiBrowsen suodatuksen käyttö

Useamman fasetin valinta muodostaa leikkauksen järjestelmän kuvista. Leikkaukseen kuuluvat Image-oliot saa kustumalla OntologyClass- luokan getIntersectHash()-metodia halutuilla kuvilla. Sen jälkeen kokoelma kuvista luodaan OntologyClass-luokan getIntersectImages()-metodilla parametreinaan edellä saatu leikkaus ja haluttu syvyys.

Esimerkki

```
Hashtable intersect = OntologyClass
    .getIntersectHash(model.getAllImages());
Vector pictures = new Vector(
    node.getIntersectImages(intersect, OntologyClass.ALL));
```

3.3.4 Kuvaan liittyvien toisten kuvien hakeminen

Kuvan ominaisuudet haetaan ensin kutsumalla Image-luokan getProperties()-metodia. Tämän jälkeen selvitetään millä ominaisuuksilla on arvoinaan instansseja. Näiltä instansseilta voi kysyä niihin liittyvät kuvat Instance-luokan getImages()-metodilla.

Esimerkki

```
Image image = new Image(...);
Vector linkedImages = new Vector();

Iterator iterator = image.getProperties().iterator();
while (iterator.hasNext()) {
    ResourceProperty property = (ResourceProperty)iterator.next();
    if(property.getValue().isInstance()) {
        Instance instance = property.getValue().getInstance();
        linkedImages.addAll(instance.getImages());
    }
}
```

3.3.5 Ontologiaresurssin ominaisuuksien haku

Ontologiaresurssiin, esimerkiksi kuvaan, liittyviä ominaisuuksia on kahdentyyppisiä. On ns. ennaltamääriteltyjä ominaisuuksia joita on jokaisella tietyntyyppisellä ominaisuudella ja muita ominaisuuksia joiden olemassaolo vaihtelee resurssin mukaan. Ennaltamääritellyillä ominaisuuksilla on omat aksessorinsa, mikäli niitten hakeminen metatietomoduulin ulkopuolelta on järkevää. Muita ominaisuuksia haetaan kutsumalla ontologiaresurssia vastavalle oliolle getProperties()-metodia. Tämä metodi palauttaa vain ne ominaisuudet jotka eivät ole ennaltamääriteltyjä.

getProperties()- metodi palauttaa ResourceCollection-olion jossa on ominaisuuksia edustavia ResourceProperty-olioita. getNamedProperty()- metodi palauttaa ResourceProperty-olion annetun nimisestä ominaisuudesta tai null jos annetun nimistä ominaisuutta ei ole olemassa.

Saadu(i)sta ominaisuuksista voi kysellä ominaisuuden arvoa getValue()- metodilla. Saadulta PropertyValue- olioilta voi kysyä arvon tyyppiä ja itse arvoa. Tarkempaa tietoa löytyy luokkn kuvauksesta luvussa 3.5.14.

Esimerkki

```
Iterator iterator = resource.getProperties().iterator();
while (iterator.hasNext()) {
    ResourceProperty property = (ResourceProperty)iterator.next();
    if(property.getValue().isInstance()) {
        Instance instanceValue = property.getValue().getInstance();
    } else if(property.getValue().isLiteral()) {
        String stringValue = property.getValue().getLiteral();
    }
}
```

3.3.6 Painojen käyttäminen

Ontologiaresursseilla on paino, jota voi lukea ja asettaa tai kasvattaa yhdellä. Painoja voidaan käyttää ontologiaresurssien järjestämiseen ja suodatinmoduuli käyttää niitä osana linkkien utiliteettia. Painot ovat long-tyyppisiä arvoja. Painoja voidaan lukea metodilla getWeight(), asettaa metodilla setWeight() ja lisätä yhdellä metodilla addWeight().

Esimerkki

```
long weight;
weight = resource.getWeight();
weight = resource.addWeight();
weight = resource.setWeight(weight * 2);
```

3.4 Moduulin sisäinen rajapinta

Luvussa kuvataan lyhyesti miten luvussa 8.2 mainitut operaatiot on toteutettu moduulin sisällä.

3.4.1 Mallin luominen ja alustaminen

MuseumModel-luokan konstruktori vain asettaa kentiin annetut url- ja path-arvot. MuseumModel-luokan loadModel()-metodi käyttää Jenana ModelLoader-luokkaa mallien luomiseksi. Malleihin luetaan tietoja toisesta tiedostosta Model-luokan read()-metodilla. loadModel()-metodi myös alustaa taustalla toimivan säieen joka on ModelSaver-luokan ilmentymä. ModelSaver-luokka on MuseumModel-luokan sisäluokka, ja se ajaa run()-metodissaan MuseumModel-olion writeModel()-metodin kerran tunnissa. writeModel() ajetaan myös säieen finalize()-metodissa, eli levylle kirjotetaan vielä MuseumModel-olion tuhoamisen yhteydessä.

3.4.2 HiBrowsen fasettipuiden hakeminen

MuseumModel-luokan getFacetRoots()-metodi hakee fasettiontologiasta urit faseteille ja hakee näillä ureilla fasettien juuria vastaavat OntologyClass-oliot perusontologiasta. Juurille haetaan alipuut kutsumalla rekursiivisesti OntologyClass-olioille getChildren()-metodia syvyydellä 0. getChildren()-metodi hakee perusontologiasta ne ontologialuokat joiden isänä on kutusuva olio. Isä-poika suhdetta kuvataan ontologiassa RDF:n subclassOf-propertylla.

3.4.3 HiBrowsen suodatuksen käyttö

Tästä ei ole mitään erityistä sanottavaa. Kutsutaan jokaiselle annetulle OntologyClass-oliolle getImages()-metodia ja leikautetaan näitten palauttamat kuvat. Leikkaus on matemaattisen joukko-opin mukainen leikkaus jossa ei ole kaksoiskappaleita. Nopeussyistä leikkaus palautetaan getIntersectHash()-metodista hajautustaulukkona, jolloin yhden fasetin tarkentuessa voidaan nopeammin päivittää jo olemassaolevaa leikkausta.

3.4.4 Kuvaan liittyvien toisten kuvien hakeminen

Kuvan 2.3 mukaan kuvakorttiin liittyy aina joukko instansseja ja ontologialuokkia. Oletetaan että kaksi kuvaa liittyy sisällöllisesti toisiinsa jos niihin liittyy vähintään yksi yhteinen instanssi. Tämä on rajoittunut oletus ja kahden kuvan välisen linkin käsite pitäisi parantaa tulevissa versioissa.

Ominaisuuksien haku ja käsittely on käsitelty alla.

3.4.5 Ontologiaresurssin ominaisuuksien haku

Ontologiaresurssiin liittyvät ominaisuudet haetaan perusontologiasta OntologyResource-luokan getProperties()-metodilla. Metodi hakee kaikki lauseet jotka liittyvät kutsuvan olion ominaisuuksiin, suodattaa niistä pois ns. ennalatamääritellyt ominaisuudet ja yrittää luoda ominaisuuden arvolle Object-olion. Ominaisuudelle luodaan myös CompositeId-olio ominaisuuden tunnisteeksi. Lopputulos palautetaan ResourceCollection-oliona.

3.4.6 Painojen käyttäminen

OntologyResource-luokan addWeight()- ja setWeight()-metodit yrittävät luoda paino-ontologiaan tarvittavat lauseet (statement) mikäli niitä ei kyseiselle resurssille löydy. Tämän lisäyksen suorittaa metodi addWeightStatements(). Jenassa lausetta edustaa StatementImpl-luokka joka on kolmikko joka koostuu kahdesta resurssista subject ja object sekä yhdestä ominaisuudesta property. Resurssia edustaa luokka ResourceImpl ja ominaisuutta luokka PropertyImpl. Kohde voi olla myös literaali jota edustaa luokka LiteralImpl. Lause luodaan luomalla ensin kolmikon osat ja kustumalla sitten PropertyImpl-luokan konstruktoria parametrinaan nämä osat. Lause lisätään ontologiaan Jenan Model-luokan add()- metodilla. Ontologiassa olevaa lausetta muokataan ensin hakemalla se StatementImpl-olioksi ja muokkaamalla se StatementImpl-luokan set()-metodilla.

3.4.7 ResourceCollectionin muodostaminen

ResourceCollectionin alaluokka StatementResourceCollection on sikäli erikoinen että tarvittava tieto haetaan Jenan avulla vasta kun se tarvitaan. Ainoa tapa päästä kokoelman alkioihin käsiksi on iteraattorilla, joka saadaan kustumalla ResourceCollection-olion iterator()-metodia. Kun StatementResourceCollection luodaan sen konstruktorille annetaan parametriksi Model-olio mallista josta lauseet haetaan, SelectorImpl-olio valitsijasta joka suodattaa mallin lauseet ja ResourceProvider-olio joka annetaan sisäluokkana. Annetun ResourceProvider-olion pitää toteuttaa getResource()-metodi joka luo parametrinaan saamastaan lauseesta halutun OntologyResource-tyyppisen olion.

3.5 Luokkakuvaukset

Seuraavaksi kuvataan metatietomoduulin luokat, sekä luokkien sisäiset kentät, metodit ja tietorakenteet.

3.5.1 Luokka MuseumModel

Tämä luokka mahdollistaa Jenan käyttämisen RDF-tiedostojen ja ontologoiden lukemiseen ja kirjoittamiseen. Luokkaan sisältyy myös taustasäie, joka ajaa writeModel()-metodin tietyin väliajoin. Taustasäie mahdollistaa klikkausmäärien muutosten kirjaamisen pysyvästi levylle.

Kentät

Tämän luokan kentät ovat pääasiassa String- ja Model-olioita, joihin on talletettu polkujen nimiä ja eri ontologioiden käsittelyyn tarkoitettuja Jenan Model-olioita.

public final static String TYPE_PROPERTY_URI = "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"

”Type”-tyyppisen RDF-ominaisuuden predikaatin tunniste. RDF-syntaksin mukainen.

public final static String LABEL_PROPERTY_URI = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#label"

”Label”-tyyppisen RDF-ominaisuuden predikaatin tunniste. RDF-syntaksin mukainen.

public final static String SUBCLASSOF_PROPERTY_URI = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#subClassOf"

”SubClassOf”-tyyppisen RDF-ominaisuuden predikaatin tunniste. Ilmaisee alaluokkasuhdetta. RDF-syntaksin mukainen.

public final static String WEIGHT_ID_PROPERTY_URI = "http://protege.stanford.edu/klik#Id"

Tunniste Protégé-ohjelmalla tehtyyn paino-ontologian, jolla tunnistetaan tietyyn ontologiaresurssiin kuuluvan painon.

public final static String WEIGHT_COUNT_PROPERTY_URI = "http://protege.stanford.edu/klik#count"

Tunniste paino-ontologian count-kenttään, eli painon arvon.

public final static String RESOURCE_CLASS_URI = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"

"Resource"-tyyppisen RDF-ominaisuuden predikaatin tunniste. RDF-syntaksin mukainen.

public final static String RESOURCE_WEIGHT_URI = "http://protege.stanford.edu/klik#resource_weight"

Paino-ontologian reource_weight kentän tunniste.

public final static String BASE_NS = "http://protege.stanford.edu/Promootio#"

Promootio-ontologian tunniste ja nimiavaruuden uri.

public final static String WEIGHT_NS = "http://protege.stanford.edu/klik#"

Paino-ontologian tunniste ja nimiavaruuden uri.

public final static String IMFILE_NS= "http://protege.stanford.edu/imagefile#"

Kuvatiedosto-ontologian tunniste ja nimiavaruuden uri.

public final static String FACETNAME_PREDICATE = "http://protege.stanford.edu/fasetit#Nimi"

Fasetti-ontologian Nimi-kentän tunniste.

public final static String INSTANCELABEL_PREDICATE = "http://protege.stanford.edu/Promootio#Nimi"

Promootio-ontologian instanssin Nimi-kentän tunniste.

public static final String NAME_PREFIX = "#"

Kenttien ja alkutunnisteen erotinmerkki.

public final static String INSTANCE_PREFIX = "promootio_"

Instanssin tunniste.

public final static String IMAGE_PREFIX = "img_"

kuvan tunniste RDF-tiedostoissa.

public final static String WEIGHT_SUFFIX = "_weight"

Painon tunniste.

public final static String IMAGE_PROTOCOL = "file://"

Tiedosto-protokollan tunniste.

public final static String DC_DESCRIPTION = "http://purl.org/metadata/dublin_core#Description"

Dublin Core-kirjastomallin mukainen kuvaus.

public final static String DC_SUBJECT =
"http://purl.org/metadata/dublin_core#Subject"

Dublin Core-kirjastomallin mukainen aihetunniste.

public final static String IMAGE_PATH =
"pictures/"

Kuvien hakemiston tunniste.

private Model baseModel

Perusontologian oliomallia vastaava Jenan Model-olio. Tässä tapauksessa tämä Model-olio kuvaa promootio-ontologiaa.

private Model facetModel

Fasettiontologian oliomallia vastaava Jenan Model-olio.

private Model weightModel

Paino-ontologian oliomallia vastaava Jenan Model-olio.

private Model imagefileModel

Kuvatiedosto-ontologian oliomallia vastaava Jenan Model-olio.

private String basePath

Hakemisto, josta tähän MuseumModel-luokkaan kuuluvat tiedostot löytyvät.

private java.net.URL baseURL

Ohjelmiston koti URL, josta löytyy kaikki selaimen tarvitsema tieto.

private boolean modelReadAlready

Kertoo, onko tähän MuseumModel-malliin luettu jo sen tarvitsemat tiedot. Estää tietojen lukemisen kahteen kertaan, sekä mallin kirjoittamisyrityksen jos mallia ei ole vielä luettu.

private Item imageWeightMaximumItem

Item-olio, joka sisältää suosituimman kuvan painon.

private Item classWeightMaximumItem

Item-olio, joka sisältää suosituimman ontologialuokan painon.

private Item imagePropertyWeightMaximumItem

Item-olio, joka sisältää suosituimman kuvaominaisuuden painon.

Konstruktorit

Tällä luokalla on vain yksi konstruktori:

public MuseumModel(java.net.URL baseURL, String basePath)

Luodaan uusi MuseumModel-olio. Saa parametreina perus URLIN `baseURL` ja perushakemiston `basePath`, jotka talletetaan niille varattuihin kenttiin. Konstruktorin sisällä asetetaan myös suosituimpien , eli klikatuimpien, kuvien yms. kenttien arvot. Esim. classWeightMaximumItem.

Metodit

Suurin osa tämän luokan metodeista käsittelee eri ontologioita:

public boolean readModel()

Luo varsinaisen MuseumModelin levyltä luettavista RDF-dokumenteista. Käyttää hyväkseen Jenan metodeja. Antaa arvot kaikille Model-päätteisille kentille. Palauttaa `true` mikäli kaikki sujui ongelmitta.

public boolean writeModel()

Kirjoittaa `weightModel`-olion ja `imagefileModel`-olion levylle. Metodi on synkronisoitu. Palauttaa `true`, jos kirjoittaminen onnistui.

public ResourceCollection getFacetRoots() throws RDFException

Hakee perusontologian fasettijuuret, jotka on määritelty fasettiontologiassa. Fasettien juuret määräävät fasetit. Palauttaa `ResourceCollection`-olion, jolle on implementoitu Javan `Collection`-rajapinta. Heittää poikkeuksen, jos haussa tapahtuu virhe.

public ResourceCollection getAllImages()

Palauttaa `ResourceCollection`-olion kaikksta kannasta löytyvistä kuvista.

public ResourceCollection getPopularFacets(int numberOfFacets)

Palauttaa `ResourceCollection`-olion, jossa on enintään numberOfFacets suosituinta, eli klikatuinta fasettia. Negatiivisella parametrilla haetaan kaikki fasetit. Metodi järjestää fasetit laskevaan järjestykseen suosituimmasta vähiten käytettyyn.

public ResourceCollection getPopularImages(int numberOfImages)

Palauttaa `ResourceCollection`-olion, jossa on enintään `numberOfImages` suosituinta kuvaa.

public OntologyClass getOntologyClass(Id classId)

Palauttaa `classId`:n mukaisen ontologialuokan, eli `OntologyClass`-olion.

public Image getImage(Id imageId)

Palauttaa `imageId`:n yksilöimän kuvan, eli `Image`-olion.

public long getImagePropertyWeightMaximum()

Palauttaa mallin suosituimman kuvaominaisuuden painon.

public long setImagePropertyWeightMaximum(long count)

Asettaa painon suosituimmalle kuvaominaisuudelle. Uudeksi arvoksi tulee `count`, mikäli vanha arvo on pienempi kuin uusi. Palauttaa maksimiarvon.

public long getImageWeightMaximum()

Palauttaa mallin suosituimman kuvan painon.

public long setImageWeightMaximum(long count)

Asettaa painon suosituimmalle kuvalle. Uudeksi arvoksi tulee `count`. Palauttaa maksimiarvon.

public long getClassWeightMaximum()

Palauttaa mallin suosituimman ontologialuokan painon.

public long setClassWeightMaximum(long count)

Asettaa painon suosituimmalle ontologialuokalle. Palauttaa maksimiarvon.

public String getBasePath()

Palauttaa mallin perushakemiston. Hakemiston alta löytyvät kuva- ja rdf-tiedostot.

public URL getBaseURL()

Palauttaa mallin perus-URLin, josta löytyy kaikki selaimen tarvitsema tiedot.

public Model getBaseModel()

Palauttaa kentän baseModel sisällön.

public Model getWeightModel()

Palauttaa kentän weightModel sisällön.

public Model getImagefileModel()

Palauttaa kentän imagefileModel sisällön.

public Model getFacetModel()

Palauttaa kentän facetModel sisällön.

public static String encodeURI(String stringToEncode)

Toteuttaa Jenan URIref.encode -metodin. Koodaa annetun Unicode Stringin URI yhteensopivaksi ja palauttaa koodatun Stringin.

public static String decodeURI(String stringToDecode)

Muuntaa annetun URI-Stringin Unicode Stringiksi ja palauttaa Unicode Stringin.

3.5.2 Luokka WeightComparator

`MuseumModel`-luokka käyttää tätä luokkaa `OntologyResource`-olioiden vertailuun näiden painon perusteella. Tämä luokka toteuttaa Javan `Comparator`-rajapinnan.

Konstruktorit

Luokalle ei ole toteutettu konstruktoreita, joten ainoa mahdollinen konstruktori on tyhjä konstruktori.

Metodit

Metodeita on toteutettu tasan yksi:

public int compare(Object o1, Object o2)

Metodi olettaa, että annetut parametrit o1 ja o2 ovat `OntologyResource`-olioita. Metodi kysyy parametriolioilta niiden painot ja muuttaa ne `Long`-olioiksi. Tämän jälkeen metodi palauttaa Javan `compareTo`-metodin tuloksen.

3.5.3 Luokka OntologyResource

`OntologyResource` on abstrakti luokka joka implementoi javan `Comparable`-rajapinnan. Se toteuttaa kaikkien ontologiaresurssien yhteiset metodit ja kentät. Eri resursseilla on huomattavan paljon yhteisiä metodeja ja kenttiä, joita ei ole järkevää toistaa jokaisessa muussa luokassa uudestaan. Näin luokkarakenne kevenee huomattavasti.

Kentät

protected final Id id

Id-muuttuja, jolla voidaan yksikäsitteisesti tunnistaa tämä ontologiaresurssi.

protected java.lang.String label

String-muuttuja, johon voidaan tallentaa tämän ontologiaresurssin nimike. Nimike on näytölle tulostettavissa oleva loppukäyttäjälle informatiivinen nimi ontologiaresurssille.

protected final MuseumModel model

MuseumModel-muuttuja, johon on tallennettu osoitin museomalliin, jonka osa tämä ontologiaresurssi on.

protected static java.util.Hashtable predefinedPropertyHash

Hajautustaulukko-muuttuja, johon on tallennettu niitten ominaisuuksien URIt jotka eivät ole annotointitietoa ja siten pitää jättää pois tämän luokan getProperties()-metodin palauttamista ominaisuuksista..

Konstruktorit

public OntologyResource(OntologyResource another)

Konstruktori, joka saa parametrikseen toisen OntologyResource-olion `another`-muuttujassa ja tekee siitä kopion.

public OntologyResource(MuseumModel model, Id id)

Konstruktori, joka saa parametrikseen MuseumModel-olion `model`-muuttujassa ja Id-olion `id`-muttujassa, ja asettaa niitten arvot luokan vastaavien `model`- ja `id`-kenttien arvoiksi.

public OntologyResource(MuseumModel model, java.lang.String label, Id id)

Konstruktori, joka saa parametrikseen MuseumModel-olion `model`-muuttujassa, nimikkeen `label`-muuttujassa ja Id-olion `id`-muttujassa, ja asettaa niitten arvot luokan vastaavien `model`-, `label`- ja `id`-kenttien arvoiksi.

Metodit

public int compareTo(Object anotherOntologyResource) throws ClassCastException

Osa javan `Comparable`-rajapintaa. Vertailee ontologiaresursseja id-kentän mukaan. Tarkempaa tietoa löytyy javan API-kuvauksen `Comparable`-rajapinnan kuvauksesta.

public boolean equals(java.lang.Object another)

Korvaa javan `Object`-luokan equals-metodin. Tarkempaa tietoa löytyy javan API-kuvauksen `Object`-luokan kuvauksesta.

protected static java.lang.String getNameForURI(java.lang.String uri)

Palauttaa `MuseumModel`-luokan decodeURI-metodilla suodatetun ontologialuokan URIn paikallisnimen.

public static java.util.Hashtable getIntersectHash(java.util.Collection collections) throws java.lang.IllegalArgumentException

Palauttaa `Hashtable`-olion hajautustaulukosta jossa on annettujen kokoelmien alkoiden leikkaus. Alkioihin on tallennettu ontologiaresursseja ja leikkaus suoritetaan käyttäen tämän luokan equals-metodia. Lopputuloksena on siis taulukko jossa on kaikki ne ontologiaresurssit jotka esiintyvät kaikissa annetuissa kokoelmissa ilman duplikaatteja. Hajautustaulukkoa käytetään tehokkaiden lisäleikkausten tekemisessä, jolloin jollekkin kokoelmalle ontologiaresursseja tarkistetaan resurssi kerralla kuuluvatko ne tiettyyn leikkaukseen.

public static ResourceCollection intersect(java.util.Hashtable intersect)

Palauttaa `ResourceCollection`-olion kokoelmasta johon sisältyy getIntersectHash-metodin palauttamat ontologiaresurssit.

public static ResourceCollection intersect(java.util.Collection collections)

Palauttaa `ResourceCollection`-olion kokoelmasta johon sisältyy ne ontologiaresurssit jotka esiintyvät kaikissa annetuissa kokoelmissa. Palautettu kokoelma ei sisällä duplikaatteja. Metodi on toteutettu kutsumalla edellistä intersect-metodia parametrinaan getIntersectHash-metodin palauttama hajautustaulukko. getIntersectHash-metodille annetaan parametriksi tälle metodille annetut kokoelmat.

protected java.lang.Comparable getLocalKey()

Palauttaa `Comparable`-rajapinnan toteuttavan avaimen tästä ontologiaresurssista. Avain on yksikäsitteinen siinä nimiavaruudessa johon ontologiaresurssi kuuluu.

protected java.lang.Comparable getKey()

Palauttaa `Comparable`-rajapinnan toteuttavan avaimen tästä ontologiaresurssista. Avain on yksikäsitteinen riippumatta nimiavaruudesta johon ontologiaresurssi kuuluu.

public java.net.URL getURL() throws java.net.MalformedURLException

Palauttaa `URL`-olion tämän ontologiaresurssin URIista. Varoitus: ei toimi odotetulla tavalla kaikille ontologiaresursseille (Itseasiassa vain ImageFile-olioille).

protected com.hp.hpl.mesa.rdf.jena.model.Resource getResource()

Palauttaa `Resource`-olion joka sisältää Jenan vastineen tälle ontologiaresurssille.

public Id getId()

Palauttaa tämän ontologiaresurssin tunniksen, joka on tallennettu olion id-kentän.

public java.util.Hashtable getPredefinedProperties()

Palautta tämän ontologiaresurssin ennaltamääritellyt ominaisuuden, jotka ovat tallennettu olion predefinedPropertiesHash-kentän.

public String getLabel()

Palautta tämän ontologiaresurssin nimikkeen, joka on tallennettu olion label-kentän.

public String getShortCode()

Palautta tämän ontologiaresurssin nimikkeen, joka on tallennettu olion label-kentän.

public ResourceCollection getProperties()

Palauttaa `ResourceCollection`-olion kokoelmasta joka sisältää `ResourceProperty`-olioita kaikista tähän ontologiaresurssiin kuuluvista ominaisuuksista, lukuunottamatta niitä jota olion getPredefinedProperties-metodi palauttaa. Ominaisuuksien haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public long setWeight(long value)

Asettaa tämän ontologiaresurssin painon annettuun long-muotoiseen arvoon. Uusi arvo palautetaan. Jos arvoa ei entisestään ole, metodi kutsuu addWeightStatements-metodia joka luo vastaavat lauseet `MuseumModel`-luokan weightModel-kenttään.

public long addWeight()

Lisää tämän ontologiaresurssin painoa yhdellä. Uusi arvo palautetaan. Jos arvoa ei entisestään ole, metodi kutsuu addWeightStatements-metodia joka luo vastaavat lauseet `MuseumModel`-luokan weightModel-kenttään.

public long getWeight()

Palauttaa tämän ontologiaresurssin painon. Jos arvoa ei ole, palautetaan 0.

private boolean addWeightStatements()

Lisää `MuseumModel`-luokan weightModel-kentän osoittamaan Jenan ontologiaan tämän ontologiaresurssin painoa kuvaavat lauseet.

private Statement getWeightValueStatement()

Palauttaa `MuseumModel`-luokan weightModel-kentän osoittamasta Jenan ontologiasta tämän ontologiaresurssin painoa kuvaavat lauseet.

public MuseumModel getModel()

Palautta `MuseumModel`-olion johon tämä ontologiaresurssi kuuluu. Malli on tallennettu olion model-kentän.

public ResourceProperty getNamedProperty(java.lang.String value, com.hp.hpl.mesa.rdf.jena.model.Model jenaModel)

Palauttaa `ResourceProperty`-olion annetulla URIlla annetusta `Model`-olion kuvaamasta Jenan oliomallista. Ominaisuuksien haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public boolean isComparable()

Osa javan `Comparable`-rajapintaa. Palauttaa toden arvon.

public int hashCode()

Palauttaa hajautustalukon indeksointiin käytettävän numeroarvon. Korvaa javan `Object`-luokan hashCode-metodin.

public java.lang.String toString()

Palauttaa `String`-olion tämän olion label-kentästä. Korvaa javan `Object`-luokan toString-metodin.

3.5.4 Luokka Item

`Item` on `OntologyResource`-luokan alaluokka ja perii sen metodit ja kentät. `Item`-olio edustaa ontologian kuvaamaa ontologiaresurssia. `Item`-luokan ilmentymillä voidaan laskea ja tallentaa painoja mille tahansa ontologiassa esiintyvälle resurssille, vaikka tälle ei olisikaan toteutettu omaa luokkaa. Esimerkkinä kahden kuvan väliselle, useamman ominaisuuden yhdistelmästä koostuvalle linkille voidaan laskea oma paino. Ainoa vaatimus on että kaikille näin käytetyille `Item`-luokan ilmentymille voidaan muodostaa oma, yksikäsitteinen tunnus, joka on jonkun `Id`-luokan alaluokan ilmentymä.

Konstruktorit

public Item(Item another)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public Item(MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

3.5.5 Luokka AbstractItem

`AbstractItem` on `Item`-luokan abstakti alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. `AbstractItem`-olio edustaa ontologian `ConcreteItem`-luokan ilmentymien kuvaamaa tosimaailman oliota. Esimerkiksi taulu voisi olla `ConcreteItem`-luokan ilmentymä. Taulussa kuvataan hedelmäasetelma. Tällöin asetelman osa, esimerkiksi päärynä, olisi tämän luokan tarkoittama `AbstractItem`. Eli `AbstractItem` kuuluu `ConcreteItem`in kuvaamaan kokonaisuuteen. Järjestelmän tässä versiossa ei ole toteutettu `AbstactItem`-luokan alaluokkia.

Konstruktorit

public AbstractItem(AbstractItem another)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public AbstractItem(MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

3.5.6 Luokka ConcreteItem

`ConcreteItem` on `Item`-luokan abstakti alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. `ConcreteItem`-olio edustaa ontologian ilmentymien kuvaamaa ontologiaresurssia. `ConcreteItem`-olio voi kuvata samalla yhtä tai useampaa `AbstractItem`-oliota. Tämä luokka kuvaa yhtä tiettyä kokonaisuutta tai asiaa, joka liittyy ontologiaan. `ConcreteItem` voi olla esim. kuva, video tai taulu, jossa on puolestaan muita ihmisiä, asioita ja esineitä eli `AbstractItem`-olioita.

Konstruktorit

public ConcreteItem(ConcreteItem another)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public ConcreteItem(MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

3.5.7 Luokka Image

`Image` on `Item`-luokan alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. `Image`-olio sisältää tietyn kuvan annotointitiedot. Eli yksi `Image`-olio vastaa yhtä oikeaa kuvaa. Kuvasta voi olla useita eri versioita, esim. GIF ja JPEG-kuvat, mutta kuva-aihe on silti sama.

Kentät

protected static Hashtable predefinedPropertyHash

Korvaa `OntologyResource`-luokan vastaavan kentän.

Konstruktorit

Luokkakonstruktori static

Korvaa `OntologyResource`-luokan vastaavan kentän. Lisää tämän luokan predefinedPropertyHash-kenttään `OntologyResource`-luokan predefinedPropertyHash-kentän sisällön sekä pelkästään `Image`-luokan ennaltamääritellyt ominaisuudet.

public Image(Image another)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public Image(MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

Metodit

public Hashtable getPredefinedProperties()

Korvaa `OntologyResource`-luokan vastaavan metodin, jotta se palauttaisi nimenomaan `Image`-luokan predefinedPropertyHash-kentän.

public String getDescription()

Palauttaa tämän kuvan nimetyn kuvausominaisuuden. Ominaisuuden URI on määritelty Dublin Core-ontologiassa ja sen arvo on määritelty `MuseumModel`-luokan DC_DESCRIPTION-vakiossa.

public String getSubject()

Palauttaa tämän kuvan nimetyn otsikko-ominaisuuden. Ominaisuuden URI on määritelty Dublin Core-ontologiassa ja sen arvo on määritelty `MuseumModel`-luokan DC_SUBJECT-vakiossa.

public String getLabel()

Palauttaa tämän kuvan getSubject-metodin palauttaman merkkijonon tai tämän kuvan getLabel-metodin palauttaman merkkijonon jos otsikkoa ei ole.

public String getShortCode()

Korvaa `OntologyResource`-luokan vastaavan metodin. Palauttaa kuvan annotoinnissa käytetyn nimikkeen ilman 'img_'-alkuosaa. Vastaa Mopsin kuvanumeroja.

public ResourceCollection getImageFile(int imageCode)

Palauttaa `ImageFile`-olion kuvatiedostosta tälle kuvalle annetulla kuvamuodolla.

public ResourceCollection getInstancesForClass(final OntologyClass oclass)

Palauttaa kokoelman niistä tähän kuvaan liittyvistä `Instance`-olioista, jotka liittyvät myös annettuun `OntologyClass`-olioon. Ontologiainstanssien haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public boolean addFileStatements(int imageCode)

Lisää `MuseumModel`-luokan getImagefileModel-metodin palauttamaan kuvatiedosto-ontologiaan lauseet tähän kuvaan littyvää annetulla kuvamuodolla tehtyä kuvatiedostoa varten.

3.5.8 Luokka ImageFile

`ImageFile` on `Image`-luokan alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. `ImageFile`-olio sisältää tietyn kuvatiedoston tiedot. Tämä luokka tarvitaan, koska yksi kuva voi olla talletettu järjestelmään useassa eri formaatissa esim. GIF tai JPEG-kuvana. Lisäksi sama kuva voi olla talletettu usealla eri resoluutiolla. Yhteen kuvaan voi siis liittyä monta kuvatiedostoa, eli `ImageFile`-oliota.

Kentät

private Image parentImage

Muuttuja johon tallennetaan sen kuvan `Image`-olio johon tämä kuvatiedosto liittyy.

protected static java.util.Hashtable predefinedPropertyHash

Korvaa `OntologyResource`-luokan vastaavan kentän.

public final static String ID = "image_id"

Tunniste jolla kuvatiedoston kuva tallennetaan kuvatiedosto-ontologiaan.

public final static String WIDTH = "width"

Tunniste jolla kuvatiedoston leveys tallennetaan kuvatiedosto-ontologiaan.

public final static String HEIGHT = "height"

Tunniste jolla kuvatiedoston korkeus tallennetaan kuvatiedosto-ontologiaan.

public final static String COLOUR = "colour"

Tunniste jolla kuvatiedoston väri tallennetaan kuvatiedosto-ontologiaan.

public final static String BASE_URL = "baseurl"

Tunniste jolla kuvatiedoston perus-URL tallennetaan kuvatiedosto-ontologiaan.

public final static String FILENAME = "filename"

Tunniste jolla kuvatiedoston tiedostonimi tallennetaan kuvatiedosto-ontologiaan.

public final static String FORMAT = "format"

Tunniste jolla kuvatiedoston tiedostomuoto tallennetaan kuvatiedosto-ontologiaan.

public final static int ORIGINAL = 0

Peruskuvan tunniste.

public final static int THUMBNAIL = 1

Esikatselukuvan tunniste.

private Image parentImage

Muuttuja johon tallennetaan sen kuvan `Image`-olio johon tämä kuvatiedosto liittyy.

Konstruktorit

Luokkakonstruktori static

Yliajaa `OntologyResource`-luokan vastaavan kentän. Lisää tämän luokan predefinedPropertyHash-kenttään `OntologyResource`-luokan predefinedPropertyHash-kentän sisällön sekä pelkästään `ImageFile`-luokan ennaltamääritellyt ominaisuudet.

public ImageFile(ImageFile another)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public ImageFile(MuseumModel model, OntologyResource parent, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria ja asettaa tämän luokan parentImage-muuttujan arvoksi parent-parametrin arvon.

Metodit

public Hashtable getPredefinedProperties()

Korvaa `OntologyResource`-luokan vastaavan metodin, jotta se palauttaisi nimenomaan `ImageFile`-luokan predefinedPropertyHash-kentän.

public String getColour()

Palauttaa tämän kuvatiedoston nimetyn väri-ominaisuuden.

public String getFileName()

Palauttaa tämän kuvatiedoston nimetyn tiedostonimi-ominaisuuden.

public String getBaseURL()

Palauttaa tämän kuvatiedoston nimetyn perus URL-ominaisuuden.

public String getFormat()

Palauttaa tämän kuvatiedoston nimetyn tiedostomuoto-ominaisuuden.

public java.net.URL getURL()

Palauttaa tämän kuvatiedoston nimetyn URL-ominaisuuden `java.net.URL`-oliona. URL koostuu kuvatiedoston perus-URLista ja sen tiedostonimestä.

public String getHeight()

Palauttaa tämän kuvatiedoston nimetyn korkeus-ominaisuuden. Käyttää tämän luokan getSize-metodia.

public String getWidth()

Palauttaa tämän kuvatiedoston nimetyn leveys-ominaisuuden. Käyttää tämän luokan getSize-metodia.

protected Dimension getSize()

Palauttaa tästä kuvatiedostosta Dimension-olion, jolta voi kysyä kuvatiedoston leveyden ja korkeuden. Tallentaa arvot kuvatiedosto-ontologiaan tulevaa käyttöä varten.

public Id getImageId()

Palauttaa tämän kuvatiedoston isäkuvan tunnuksen.

3.5.9 Luokka OntologyElement

`OntologyElement` on `OntologyResource`-luokan abstrakti alaluokka ja perii sen metodit ja kentät. `OntologyElement`-olio edustaa ontologiaresurssia, joka ei ole ontologian kuvaama resurssi. Tämän luokan suoria alaluokkia ovat `OntologyClass` ja `Instance`. Tässä luokassa ei ole toteutettu yhtäkään metodia, mutta luokka onkin suunniteltu yleiskäyttöiseksi. Metodien lisäämiselle saattaa tulla tarvetta tulevaisuudessa.

Konstruktorit

public OntologyElement(OntologyElement another)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public OntologyElement(MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

3.5.10 Luokka OntologyClass

`OntologyClass` on `OntologyElement`-luokan alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. Tämän luokan ilmentymä kuvaa yhtä ontologian solmua. Esimerkiksi Juhlasali on eräs käyttämämme Promootio-ontologian `OntologyClass` eli ontologiasolmu.

Kentät

public static final int ALL

Vakio jota käytetään syvyysarvona kutsuttaessa niitä metodeja jotka sen tarvitsevat. ALLin arvo on -1, ja vastaa aina loppumatonta syvyyttä.

public static final int DIRECT

Vakio jota käytetään syvyysarvona kutsuttaessa niitä metodeja jotka sen tarvitsevat. DIRECTin arvo on 0.

private static Hashtable imageCountHash

Hajautustaulukko-luokkamuuttuja, jota käytetään kuvien määrien muistamiseen. Taulukolla tehostetaan määräarvojen toistuvaa hakemista.

Konstruktorit

Luokkakonstruktori static

Luo `Hashtable`-olion imageCountHash-hajautustaulukolle.

public OntologyClass(OntologyClass another)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria ja asettaa erikseen luokan label-kentän.

public OntologyResource(MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria ja asettaa erikseen luokan label-kentän.

Metodit

public ResourceCollection getChildren(int depth)

Palauttaa tämän ontologialuokan depth-syvyisen alipuun lapset. Mikäli syvydeksi annetaan ALL, palautetaan koko alipuun lapset. Mikäli syvyydeksi annetaan DIRECT, palautetaan vain suorat lapset. Lapsien haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public ResourceCollection getParents(int depth)

Palauttaa tämän ontologialuokan enintään etäisyydellä depth olevat esi-isät. Mikäli syvydeksi annetaan ALL, palautetaan kaikki esi-isät. Mikäli syvyydeksi annetaan DIRECT, palautetaan vain suorat vanhemmat. Esi-isien haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public ResourceCollection getInstances(int depth)

Palauttaa kaikki tämän ontologialuokan suorien instanssien sekä getChildren-metodin depth-parametrilla palauttamien `OntologyClass`-olioiden suorien instanssien leikkauksen. Leikkaukseen käytetään `OntologyResource`-luokan intersect-metodia. Instanssien haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public ResourceCollection getImages(int depth)

Palauttaa kaikki tämän luokan getInstances-metodin depth-parametrillä palauttamien `Instance`-olioiden getImages-metodien `Image`-luokkien leikkauksen. Leikkaukseen käytetään `OntologyResource`-luokan intersect-metodia.

public ResourceCollection getIntersectImages (Hashtable intersectImages, int depth)

Palauttaa tämän luokan getImages-metodin depth-parametrillä palauttamien `Image`-olioiden ja annetun hajautustaulukon sisältämien `Image`-olioiden leikkauksen. Leikkaukseen käytetään `OntologyResource`-luokan intersect-metodia.

public long getImageCount()

Palauttaa tähän ontologialuokkaan liittyvien kuvien määrän. Määrän hakeminen tehostetaan tallentamalla jo haetut arvot imageCountHash-hajautustaulukkoon, joka on luokkamuuttuja tässä luokassa. Uudelleen haettaessa ne löydetään nopeasti ilman ontologiaan kohdistuvia hakuja.

3.5.11 Luokka Instance

`Instance` on `OntologyElement`-luokan alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. Instance kuvaa ontologiainstanssia joka on ontologialuokan ilmentymä. Ontologiainstanssi liittää kuvat ontologialuokkiin.

Esimerkki: Kuvassa on promootiolaiva ja tohtorinhattu. Niinpä tehdään instanssit tohorinhattua vastaavasta ontologialuokasta ja myös promootiopaikasta eli laivasta.

Kentät

protected OntologyClass type

Muutuja johon tallennetaan tämän ontologiailmentymän ontologialuokan. `OntologyClass`-olio.

private static Hashtable imageCountHash

Hajautustaulukko-luokkamuuttuja, jota käytetään tähän ontologiailmentymään liittyvien kuvien määrien muistamiseen. Taulukolla tehostetaan määräarvojen toistuvaa hakemista.

Konstruktorit

Luokkakonstruktori static

Luo `Hashtable`-olion imageCountHash-hajautustaulukolle.

public Instance (MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public Instance (MuseumModel model, OntologyClass type, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria ja asettaa tämän lisäksi kentän type arvoksi annettu ontologialuokka.

Metodit

public OntologyClass getType()

Palauttaa tämän ontologiainstanssin tyyppi, joka on tallennettu olion type-kenttän. Mikäli tämän ontologiainstanssin tyyppi ei ole vielä tiedossa, se etsitään ja talletetaan type-kenttään. Tyypin haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public ResourceCollection getImages()

Palauttaa tähän ontologiainstanssiin liittyviä kuvia vastaavat `Image`-oliot. `OntologyClass`-luokan getImages-metodi kutsuu tätä metodia. Kuvien haku on toteutettu `ResourceProvider`- ja `StatementResourceCollection`-luokilla.

public long getImageCount()

Palauttaa tähän ontologiainstanssiin liittyvien kuvien määrän. Määrän hakeminen tehostetaan tallentamalla jo haetut arvot imageCountHash-hajautustaulukkoon, joka on luokkamuuttuja tässä luokassa. Uudelleen haettaessa ne löydetään nopeasti ilman ontologiaan kohdistuvia hakuja.

public java.lang.String getLabel()

Korvaa `OntologyResource`-luokan vastaavan metodin. Palauttaa tämän ontologiailmentymän nimetyn ominaisuuden arvon. Nimetyn ominaisuuden nimi on tallennettu `MuseumModel`-luokassa vakioon INSTANCELABEL_PREDICATE.

3.5.12 Luokka PropertyDefinition

`PropertyDefinition` on `OntologyClass`-luokan alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. Edustaa RDFS-määritelmän mukaista rdf:Property-tyyppistä resurssia.

Kentät

protected OntologyClass domain

Ontologialuokka johon tämä ominaisuus voi kuulua.

protected OntologyClass range

Ontologialuokka johon tämä ominaisuuden arvo voi kuulua.

Konstruktorit

public PropertyDefinition(MuseumModel model, Id id)

Konstruktori, joka kutsuu `OntologyResourcen` vastaavaa konstruktoria.

public PropertyDefinition(MuseumModel model, OntologyClass range, OntologyClass domain, Id id)

Konstruktori, joka kutsuu edellisessä konstruktorissa kutsuttua `OntologyResourcen` konstruktoria ja asettaa lisäksi domain-, ja range-kenttien arvoiksi parametreina annetut arvot.

Metodit

public OntologyClass getDomain()

Palauttaa tämän luokan domain-kentän arvon.

public OntologyClass getRange()

Palauttaa tämän luokan domain-kentän arvon.

3.5.13 Luokka ResourceProperty

`ResourceProperty` on `PropertyDefinition`-luokan alaluokka ja perii sen sekä `OntologyResource`-luokan metodit ja kentät. Edustaa ontologiaresurssin ominaisuutta. Ominaisuudelta voidaan kysyä sen arvoa.

Kentät

protected Object value

Object-muuttuja johon on tallennettu tämän ominaisuuden arvo.

private String label

String-muuttuja johon on tallennettu tämän ominaisuuden nimike.

Konstruktorit

public ResourceProperty(PropertyDefinition property, OntologyResource resource, Object value, MuseumModel model)

Konstruktori, joka kutsuu `OntologyResource`-luokan konstuktoria parametrinaan annettu model-parametri sekä `CompositeId`-olion jonka osatunnuksina on annettujen property- ja resource-parametrien tunnukset. Lisäksi konstruktori asettaa tämän olion value-kentän arvoksi value-parametrissa annetun olion.

Metodit

public String getLabel()

Palauttaa tämän luokan label-kentän arvon.

public PropertyValue getValue()

Palauttaa tämän luokan value-kentän arvon `PropertyValue`-luokan ilmentymänä.

3.5.14 Luokka PropertyValue

Luokka ominaisuuden arvon käsittelemiseen. Tämä erillinen luokka tarvitaan, koska arvo voi olla `OntologyResource`-olio.

Kentät

public final static Class LITERAL = String.class

LITERAL-arvon luokan tyyppi Javan metodin mukaisesti.

public final static Class INSTANCE = Instance.class

INSTANCE-arvon luokan tyyppi Javan metodin mukaisesti.

public final static Class INTEGER = Integer.class

INTEGER-arvon luokan tyyppi Javan metodin mukaisesti.

protected Object value

Tämän ominaisuuden arvo.

protected Class type

Tämän ominaisuuden arvon tyyppi.

protected boolean isBag

Kertoo, onko tämä ominaisuuden arvo RDF-syntaksin mukaan Bag-muotoa, eli sisältääkö arvo monia arvoja.

Konstruktorit

Luokalla on yksi konstruktori:

**public PropertyValue(Object value, MuseumModel model)
throws ClassNotFoundException**

Parametrina saadaan ominaisuuden arvo ja `MuseumModel` johon ominaisuus kuuluu. ClassNotFoundException heitetään, jos parametrin `value` `getClass()`-metodi heittää poikkeuksen.

Metodit

Tämän luokan metodit tekevät yksinkertaisia tarkasteluja ominaisuuden arvolle.

public Instance getInstance() throws IllegalStateException

Palauttaa ominaisuuden instanssiarvon tai `null`, jos arvo ei ole instanssi.

public int getInt() throws IllegalStateException

Palauttaa ominaisuuden kokonaislukuarvon tai poikkeuksen, jos arvo ei ole kokonaisluku.

public String getLiteral()

Palauttaa ominaisuuden tekstiarvon tai null, jos arvo ei ole String-muotoa.

public Class getType()

Palauttaa ominaisuuden arvon tyypin, eli mikä olio arvo on.

public long getWeight()

Korvaa samannimisen `OntologyResource`-luokan metodin, koska `PropertyValue`-oliolle ei ole järkevää laskea painoa. Tämä metodi heittää automaattisesti poikkeuksen.

public Object getValue()

Palauttaa tämän ominaisuuden arvon.

public boolean isComparable()

Palauttaa true, jos ominaisuuden arvo toteuttaa Comparable rajapinnan.

public boolean isBag()

True jos ominaisuuden arvo on RDF Bag-tyyppinen.

public boolean isInstance()

True, jos ominaisuuden arvo on instanssi.

public boolean isInt()

True, jos ominaisuuden arvo on kokonaisluku.

public boolean isLiteral()

True, jos ominaisuuden arvo on literaali.

public long addWeight()

Korvaa samannimisen `OntologyResource`-luokan metodin, koska `PropertyValue`-oliolle ei ole järkevää laskea painoa. Tämä metodi heittää automaattisesti poikkeuksen.

3.5.15 Luokka ResourceCollection

`ResourceCollection` on javan `AbstractCollection`-luokan alaluokka ja perii sen metodit ja kentät. Ainoastaan korvatut metodit toimivat, muut aiheuttavat OperationNotSupportedException-poikkeuksen. Luokka on tarkoitettu resurssien käytännölliseen käsittelyyn. Ei ole järkevää palauttaa kerrallaan vain yhtä resurssia, vaan palautetaan samalla kaikki halutun kaltaiset resurssit `ResourceCollection`-oliossa.

Kentät

private ResourceCollection tail

Muuttuja johon tallennetaan kokoelman jälkeen tuleva `ResourceCollection`-olio. `ResourceCollection`-olioista voidaan siis muodostaa ketjutettu rakenne, ja `ResourceCollection`-luokan metodit tukevat ketjussa eteenpäin siirtymistä ns. läpinäkyvästi.

Konstruktorit

protected ResourceCollection()

Konstruktori, joka luo tyhjän `ResourceCollection`-olion.

Metodit

public Collection getCollection()

Luo ja palauttaa kokoelman (tässä `LinkedList`-olio) jota tämä `ResourceCollection`-olio käyttää sisäisenä tietorakenteenaan.

public boolean addAll(Collection resourceCollection)

Toteuttaa javan `Collection`-luokan addAll-metodin. Lisää tähän `ResourceCollection`-olioon toisen `ResourceCollection`-olion sisällön ketjuttamalla.

protected abstract ResourceIterator getIterator()

Abstrakti metodi `ResourceCollection`-luokan alaluokille toteutettavaksi. Palauttaa `Iterator`-olion tämän olion sisältämistä `ResourceCollection`-olioista.

public Iterator iterator()

Toteuttaa javan `Collection`-luokan iterator-metodin. Palauttaa `Iterator`-olion tämän kokoelman ja siihen ketjutettujen kokoelmien sisältämistä `ResourceCollection`-olioista getIterator-metodia käyttäen.

public int size()

Toteuttaa javan `Collection`-luokan size-metodin. Palauttaa tämän kokoelman alkioiden määrän.

3.5.16 Luokka SimpleResourceCollection

`SimpleResourceCollection` on `ResourceCollection`-luokan alaluokka ja perii sen metodit ja kentät. `SimpleResourceCollection`-olio toteuttaa `ResourceCollection`-rajapinnan tavalliselle `OntologyResource`-olioita sisältävälle `Collection`-tyyppiselle oliolle.

Kentät

Collection resourceCollection

Muuttuja johon on tallennettu tämän kokoelman alkiot sisältävä `Collection`-olio.

Konstruktorit

public SimpleResourceCollection(Collection resourceCollection)

Konstruktori, joka luo `SimpleResourceCollection`-olion, ja asettaa sen resourceCollection-kentän arvoksi annetun `Collection`-olion.

Metodit

public boolean add(Object o) throws ClassCastException, UnsupportedOperationException, NullPointerException, IllegalArgumentException

Lisää annetun olion tähän kokoelmaan.

protected ResourceIterator getIterator()

Palauttaa `ResourceIterator`-olion iteraattorista joka sisältää tämän kokoelman alkiot.

3.5.17 Luokka StatementResourceCollection

`StatementResourceCollection` on `ResourceCollection`-luokan alaluokka ja perii sen metodit ja kentät. `StatementResourceCollection`-olio toteuttaa `ResourceCollection`-rajapinnan annetulle Jenan `Selector`-oliolle.

Kentät

protected Model model

Muuttuja johon on tallennettu se Jena-malli, jonka lauseita suodatetaan annetulla `Selector`-oliolla.

protected Selector selector

Muuttuja johon on tallennettu `Selector`-olio jolla model-kentän lauseita suodatetaan.

protected ResourceProvider resourceProvider

Muuttuja johon on tallennettu `ResourceProvider`-olio jolla model-kentän suodatettuja lauseita muutetaan `OntologyResource`-olioiksi.

Konstruktorit

public StatementResourceCollection()

Konstruktori, joka luo tyhjän `StatementResourceCollection`-olion.

public StatementResourceCollection(Model model, Selector selector, ResourceProvider resourceProvider)

Konstruktori, joka luo `StatementResourceCollection`-olion, ja asettaa sen kenttien arvoiksi parametreina annetut, samannimisten muuttujien arvot.

Metodit

protected ResourceIterator getIterator()

Palauttaa `ResourceIterator`-olion iteraattorista joka sisältää tämän kokoelman alkiot.

3.5.18 Luokka ResourceIterator

`ResourceIterator` toteuttaa javan `Iterator`-rajapinnan ja perii sen metodit ja kentät. Ainoastaan yliajetut metodit toimivat, muut aiheuttavat OperationNotSupportedException-poikkeuksen. Luokka on tarkoitettu `OntologyResource`-tyyppisten olioiden iteroimiseen. Eli oliot käydään yksi kerrallaan läpi. Iteraattorissa ei enää tehdä suodatusta, sillä tässä vaiheessa mukana ovat enää halutun kaltaiset `OntologyResource`-oliot.

Kentät

private OntologyResource next

Muuttuja joka osoittaa seuraavaan ontologiaresurssiin tässä iteraattorissa.

private ResourceIterator tail

Muuttuja joka osoittaa seuraavaan iteraattoriin, josta jatkaa kun tämän iteraattorin kaikki alkiot on käyty läpi. Toteuttaa iteraattoreiden ketjutuksen.

private ResourceProvider resourceProvider

Muuttuja johon on tallennettu Jenasta `ResourceCollection`-olioita hakeva `ResourceProvider`-olioita.

private boolean closed

Muuttuja, joka kertoo onko iteraattori sujettu. Alkuarvoksi on asetettu false.

Konstruktorit

protected ResourceIterator()

Konstruktori, joka luo uuden tyhjän `ResourceIterator`-olion.

protected ResourceIterator(ResourceProvider resourceProvider)

Konstruktori, joka luo uuden tyhjän `ResourceIterator`-olion, ja asettaa sen resourceProvider-kentän arvoksi parametrina annettu `ResourceProvider`-olio.

Metodit

public void addIterator(ResourceIterator resourceIterator)

Ketjuttaa tämän iteraattorin jälkeen annetun `ResourceIterator`-olion.

public boolean hasNext()

Palautaa toden arvon jos iteraattorin next-kenttä osoittaa `OntologyResource`-olioon. Jos ei, metodi yrittää hakea `OntologyResource`-oliota eteenpäin iteraattoriketjussa. Mikäli iteraattoriketju loppuu (kenttä tail = null) ennen kun ontologiaresurssi löydetään, palautetaan false. Toteuttaa java `Iterator`-luokan rajapintaa.

public Object next()

Palauttaa sen olion johon iteraattorin next-kenttä osoittaa. Jos seuraavaa oliota ei löydy, metodi sulkee iteraattorin. Käyttää hasNext-metodia. Toteuttaa java `Iterator`-luokan rajapintaa.

protected abstract Object getNext() throws Exception

Palauttaa seuraavan olion iteraattorista. Jätetty alaluokkien toteutettavaksi.

protected abstract boolean hasNextHead() throws Exception

Palauttaa toden arvon jos iteraattorilla on seuraavaa oliota. Jätetty alaluokkien toteutettavaksi.

protected abstract void closeHead() throws Exception

Sulkee tämän iteraattorin ja vapauttaa sen varaamat resurssit. Erityisen tärkeä tietokantoja käsitellessä (tulevaisuuden asioita..). Jätetty alaluokkien toteutettavaksi.

public void close()

Kutsuu seuraavaa close-metodia false-arvoisella parametrilla.

private void close(boolean onlyThis)

Sulkee tämän iteraattorin. Jos annettun parametrin arvo on false, suljetaan koko iteraattoriketju rekursiivisesti.

protected void finalize()

Kutsuu tämän luokan close()-metodia.

public void remove() throws UnsupportedOperationException, IllegalStateException

Javan `Iterator`-rajapinnan remove()-metodia ei ole toteutettu.

3.5.19 Luokka SimpleResourceIterator

`SimpleResourceIterator` on `ResourceIterator`-luokan alaluokka ja perii sen metodit ja kentät. Luokka on tarkoitettu `OntologyResource`-tyyppisten olioiden iteroimiseen.

Kentät

private Iterator iterator

Muuttuja joka sisältää varsinaisena tietovarastona toimivan iteraattorin.

Konstruktorit

public SimpleResourceIterator()

Konstruktori, joka luo uuden tyhjän `SimpleResourceIterator`-olion.

public SimpleResourceIterator(Iterator iterator)

Konstruktori, joka luo uuden `ResourceIterator`-olion, ja asettaa sen sisällöksi annetun `Iterator`-olion sisällön.

Metodit

protected void closeHead()

Korvaa `ResourceIterator`-luokan vastaavan metodin. Ei tee mitään.

protected boolean hasNextHead()

Palauttaa toden arvon jos iteraattorilla on seuraavaa oliota.

protected abstract Object getNext()

Palauttaa seuraavan olion iteraattorista.

3.5.20 Luokka StatementResourceIterator

`StatementResourceIterator` on `ResourceIterator`-luokan alaluokka ja perii sen metodit ja kentät. Luokka on tarkoitettu `OntologyResource`-tyyppisten olioiden iteroimiseen. Luokka toimii liittymänä Jenan `StmtIterator`-luokalle.

`StatementResourceIterator`-luokka hakee iteroinnin yhteydessä ontologia-resurssit ontologiasta.

Kentät

private StmtIterator iterator

Muuttuja joka sisältää varsinaisena tietovarastona toimivan iteraattorin.

Konstruktorit

public StatementResourceIterator(StmtIterator iterator, ResourceProvider resourceProvider)

Konstruktori, joka luo uuden `StatementResourceIterator`-olion ja asettaa sen iterator- ja resourceProvider- kenttien arvoiksi parametreina annetut arvot.

Metodit

protected void closeHead()

Kutsuu iterator-kentän sisältämän `StmtIterator`-olion close-metodia.

protected boolean hasNextHead()

Palauttaa toden arvon jos iteraattorilla on seuraavaa oliota.

protected Object getNext()

Palauttaa seuraavan olion iteraattorista.

3.5.21 Luokka Id

`Id` on `Comparable`-rajapinnan alarajapintaluokka ja perii sen metodit ja kentät. Jokaisella ontologiaresurssilla on tunnuksenaan `Id`-tyyppinen olio. Kuten luokan nimestäkin voi päätellä, `Id` on jokaisella ontologiaresurrsilla yksilöllinen: kahta samanlaista ei löydy.

Metodit

public String getId()

Palauttaa merkkijonoesityksen tästä tunnuksesta.

public String getString()

Palauttaa merkkijonoesityksen tästä tunnuksesta.

3.5.22 Luokka SimpleId

`SimpleId` on `Id`-rajapinnan toteuttava luokka ja perii sen metodit ja kentät. Se käytetään `OntologyResource` olion tunnuksena silloin kun ontologiaresurssin URI on yksikäsitteinen järjestelmässä. URI on yksikäsitteinen, kun se liittyy esim. johonkin URLiin. www.cs.helsinki.fi/group/wwwmuseo kelpaisi siis `SimpleId`:ksi.

Kentät

protected String id

Muuttuja johon on tallennettu tunnus merkkijonona.

Konstruktorit

public SimpleId(String uri)

Konstruktori joka luo uuden `SimpleId`-olion ja asettaa sen tunnukseksi parametrissa annetun merkkijonon.

Metodit

public String getId()

Palauttaa merkkijonoesityksen tästä tunnuksesta.

public String getString()

Palauttaa merkkijonoesityksen tästä tunnuksesta.

public boolean equals(Object anotherId)

Palauttaa toden arvon jos annettu olio on SimpleId-olio jonka getId()-metodin palauttama merkkijono on sama kuin tämän olion id-kenttä.

public int compareTo(Object anotherId) throws ClassCastException

Toteuttaa javan `Comparable`-rajapinnan vertailemalla tunnuksien getId()-metodien palauttamia merkkijonotunnuksia.

public String toString()

Palauttaa merkkijonoesityksen tästä tunnuksesta.

3.5.23 Luokka CompositeId

`CompositeId` on `Id`-rajapinnan toteuttava luokka ja perii sen metodit ja kentät. Se koostuu kahdesta tai useammasta `Id`-tyyppisestä oliosta ja sitä käytetään `OntologyResource` olion tunnuksena silloin kun ontologiaresurssin oma URI ei ole yksikäsitteinen järjestelmässä.

Esimerkki: Erityyppisillä ontologiaresursseilla voi olla samanniminen ominaisuus. Tällöin ominaisuudesta ja sen omistajasta muodostuu yksikäsitteinen tunniste ominaisuudelle.

Kentät

protected Collection ids

Muuttuja johon on tallennettu kaikki tämän tunnusyhdisteen osatunnuksia vastaavat `Id`-tyyppiset oliot.

public final String ID_SEPARATOR = ";"

Vakio johon on tallennettu tämän tunnusyhdisteen merkkijonoesityksessä käytettävää osatunnusten erotinmerkkiä.

Konstruktorit

public CompositeId(Collection ids)

Konstruktori joka luo uuden `CompositeId`-olion ja asettaa sen tunnukseksi parametrissa annetun osatunnuskokoelman.

public CompositeId(Id id1, Id id2)

Konstruktori joka luo uuden `CompositeId`-olion ja asettaa sen tunnukseksi parametrissa annettujen tunnusten yhdisteen. Käytännössä se siis lisää annetut tunnukset ids-kentän kokoelmaan.

Metodit

public String getId()

Palauttaa merkkijonoesityksen tästä tunnuksesta. Metodi jakaa rekursiivisesti ids-kokoelmassa olevat tunnukset kunnes ne ovat yksinkertaisempia `SimpleId` ja katenoi tämän jälkeen näitten getId()-metodilla palauttaman merkkijonot erotettuina ID_SEPARATOR-vakiolla.

public String getString()

Palauttaa merkkijonoesityksen tästä tunnuksesta. Katso getId()-metodi.

public boolean equals(Object anotherId)

Palauttaa toden arvon jos annettu olio on CompositeId-olio jonka getId()-metodin palauttama merkkijono on sama kuin tämän olion id-kenttä.

public int compareTo(Object anotherId) throws ClassCastException

Toteuttaa javan `Comparable`-rajapinnan vertailemalla tunnuksien getId()-metodien palauttamia merkkijonotunnuksia.

public String toString()

Palauttaa merkkijonoesityksen tästä tunnuksesta. Katso getId()-metodi.

4 Suodatinmoduuli (TR, EM)

Suodatinmoduulin tarkoituksena on nimensä mukaisesti suodattaa tietoa, sekä järjestellä että arvottaa sen mielenkiintoarvon mukaan.

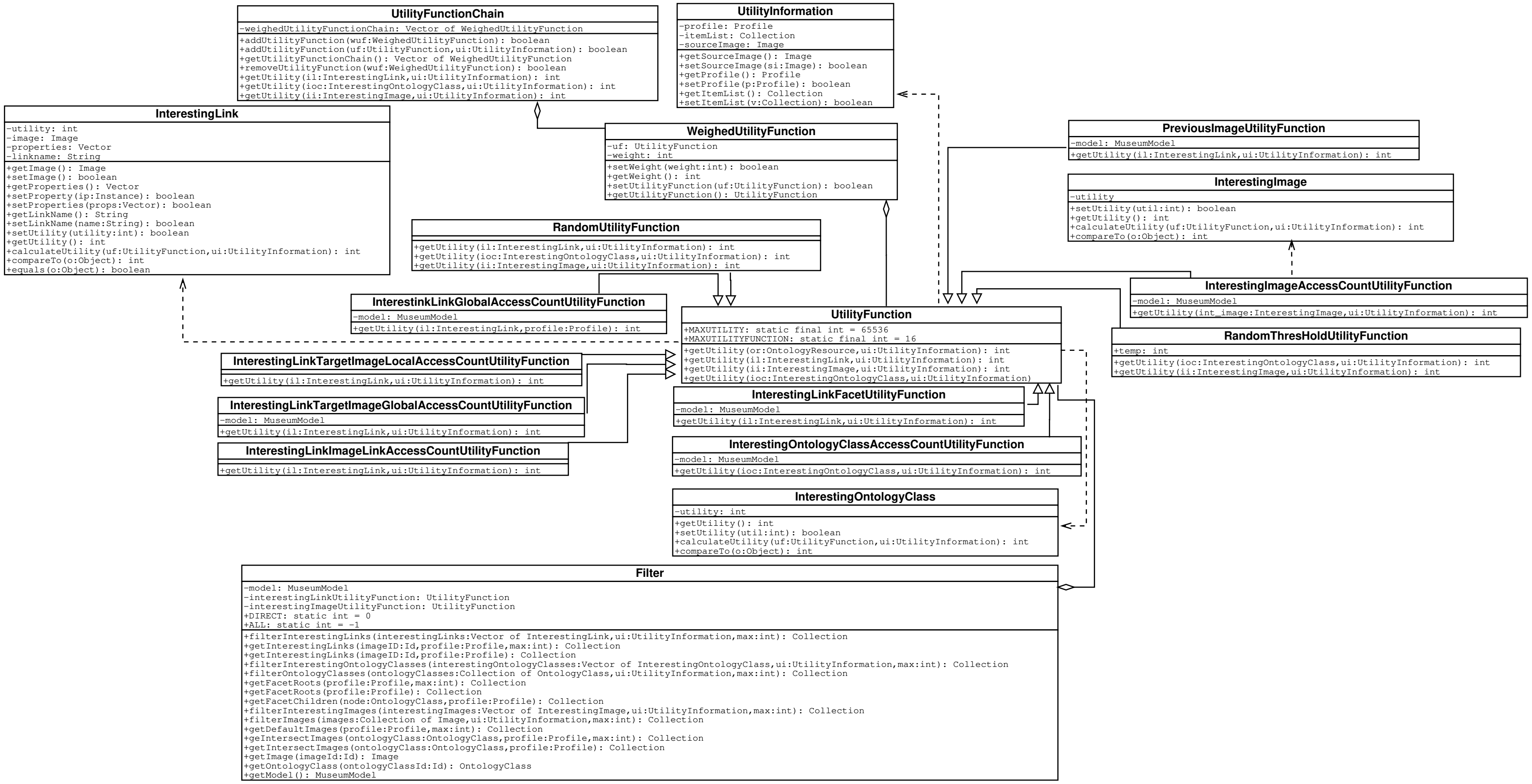
4.1 Yleiskuvaus

Suodatinmoduuli tarjoaa muille moduuleille mahdollisuuden mitata erilaisten ontologiaresurssien hyödyllisyyttä, utiliteettia, sekä näiden utiliteettien käytön resurssikokoelmien järjestämiseen ja suodattamiseen.

Resurssien utiliteettien laskemiseen tarjotaan useita erilaisia yhdistettäviä utiliteettifunktioita, joista jokainen tarkastelee resurssin hyödyllisyyttä yhdeltä rajatulta pohjalta. Resurssin koko utiliteettina käytetään osautiliteettifunktioiden painotettua keskiarvoa.

4.2 Moduulin ulkoinen rajapinta

Suodatin-moduulin rajapinta toimii välittäjänä metatietomoduulin ja käyttöliittymän välillä. Rajapinnan muodostavat luokka Filter, joka toteuttaa kaikki rajapintametodit, sekä palautustyyppiluokat InterestingLink, InterestingImage ja InterestingOntologyClass, joista kaksi viimeistä enkapsuloivat suoraan metadata-



Kuva 4: Suodatin-moduulin luokkakaavio

moduulin vastaavat oliot ja ensimmäinen sisältää kaiken mielenkiintoisten linkkien näyttämiseen tarvittavan informaation.

4.3 Moduulin sisäinen rajapinta

Moduulin toiminnan kannalta tärkeimmät operaatiot toteutetaan suoraan ulkoisessa rajapintaluokassa Filterissä. Moduulin sisälle jäävät tämän rajapinnan käyttämät erilaiset utiliteettifunktiot, sekä niistä muodostettavat painotetut ketjut.

4.4 Luokka Filter

Tämä luokka etsii Image- ja OntologyClass-olioihin liittyvät mielenkiintoiset linkit eli InterestingLink-oliot ja järjestää UtilityFunction-olion tarjoaman utiliteetin mukaan. Lisäksi luokassa on myös metodeja kuvakokoelmien (eng. Collection) järjestämiseen ilman suodatusta ja OntologyClass-olioiden ja profiilien (Profile) leikkausjoukkojen laskemiseen.

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private MuseumModel model

MuseumModel-objekti, jota käytetään esimerkiksi kuvien ja fasetin juurien hakemiseen.

private UtilityFunction interestingLinkUtility

Utiliteettifunktio, jota käytetään utiliteetin laskemiseen InterestingLink-olioille.

private UtilityFunction interestingImageUtilityFunction

Utiliteettifunktio, jota käytetään utiliteetin laskemiseen InterestingImage-olioille.

private UtilityFunction interestingOntologyClassUtilityFunction

Utiliteettifunktio, jota käytetään utiliteetin laskemiseen InterestingOntologyClass-olioille.

public static final int DIRECT=0

Vakio, jota käytetään metodikutsuissa kun halutaan solmun välittömät vanhemmat/lapset.

public static final int ALL=-1

Vakio, jota käytetään metodikutsuissa kun halutaan solmun kaikki vanhemmat/lapset aina juureen/lehteen asti.

Konstruktorit

Tällä luokallla on kaksi konstruktoria.

public Filter(MuseumModel model)

Luo Filter-olion asettaen sen model-muuttujaan saadun MuseumModel-olion.

public Filter(MuseumModel model, UtilityFunction ilu, UtilityFunctio iiu, UtilityFunction ioc)

Luo Filter-olion asettaen sen model-muuttujaan saadun MuseumModel-olion, interestingLinkUtilityFunction-muuttujaan saadun ilu-objektin, interestingImageUtilityFunction-muuttujaan saadun iiu-objektiin ja interestingOntologyClass-muuttujaan saadun iocu-objektin.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public Collection filterInterestingLinks(Vector interestingLinks, UtilityInformation ui, int max)

Järjestää InterestingLinkeistä koostuvan Vector-olion kiinnostavuuden mukaan käyttäen kutsujan InterestingLinkUtilityFunction-objektia.Palauttaa korkeintaan max kappaletta InterestingLink-olioita.

public CollectiongetInterestingLinks(Id imageID, Profile profile, int max)

Etsii annetun kuva (määritelty imageID:llä) linkit ja järjestää ne utiliteetin mukaan ja palauttaa niitä korkeintaan max kappaletta.

public Collection getInterestingLinks(Id imageID, Profile profile)

Etsii annetun kuva (määritelty imageID:llä) linkit ja järjestää ne utiliteetin mukaan.

public Collection filterInterestingOntologyClasses(Vector interestingOntologyClasses,UtilityInformation ui, int max)

Järjestää InterestingOntologyClasseista koostuva Vector-olion niiden kiinnostavuuden mukaan käyttäen utiliteettifunktiona kutsujan interestingOntologyClassUtilityFunction-muuttujaa.

public Collection filterInterestingOntologyClasses(Collection OntologyClasses, UtilityInformation ui, int max)

Toimii muuten samoin kuin edellä mainittu samanniminen metodi, paitsi OntologyClass-oliot annetaan Vector-oliossa.

public Collection getFacetRoots(Profile profile,int max)

Palauttaa annetun profiilin fasetin juuret kiinnostavuuden mukaan järjestettynä. Palauttaa enintään max kappaletta juuria.

public Collection getFacetRoots(Profile profile)

Palauttaa annetun profiilin fasetin kaikki juuret kiinnostavuuden mukaan järjestettynä.

public Collection getFacetChildren(OntologyClass node,Profile profile)

Palauttaa fasettisolmun lapset.

public Collection filterInterestingImages(Vector interestingImages,UtilityInformation ui,int max)

Järjestää interestingImage-olioista koostuvan Vectorin niiden kiinnostuvuuden mukaan, käyttäen utiliteettifunktionaan kutsujan interestingImageUtilityFunction-muuttujaa.

public Collection filterImages(Collection images,UtilityInformation ui,int max)

Järjestää saadun Image-olioista koostuvan Collectionin näiden kiinnostavuuden mukaan. Palauttaa enintään max kappaletta kuvia.

public Collection getDefaultImages(Profile profile, int max)

Palauttaa kutsujalle oletuskuvat kiinnostavuuden mukaan järjestettynä. Palauttaa enintään max kappaletta kuvia.

public Collection getIntersectImages(OntologyClass ontologyClass,Profile profile)

Palauttaa kuvat jotka liittyvät jokaiseen HiBrowse-valintaan.

public Collection getIntersectImages(OntologyClass ontologyClass,Profile profile, int max)

Palauttaa kuvat jotka liittyvät jokaiseen HiBrowse-valintaan. Palauttaa kuvia enintään syvyydestä, jonka määrittelee parametri max.

public boolean setModel(MuseumModel model)

Asettaa saadun MuseumModel-olion Filter-olion model-muuttujaksi.

public MuseumModel getModel()

Palauttaa Filter-olion model-muuttujan.

4.5 Luokka UtilityFunction

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public static final int MAXUTILITY=65536

Vakio, joka määrittelee kaikkien utiliteettifunktioiden maksimiarvon.

public static final int MAXUTILITYSHIFT=16

Vakio, jota käytetään utiliteettifunktioiden arvojen muuntamiseen.

Konstruktorit

Tällä luokallla ei ole konstruktoreita.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public int getUtility(OntologyResource or, UtilityInformation ui)

Palauttaa utiliteettifunktion arvon parametrina saadusta OntologyResource-oliosta.

public getUtility(InterestingLink il, UtilityInformation ui)

Palauttaa utiliteettifunktion arvon parametrina saadusta InterestingLink-oliosta.

public getUtility(InterestingImage ii, UtilityInformation ui)

Palauttaa utiliteettifunktion arvon parametrina saadusta InterestingImage-oliosta.

public getUtility(InterestingOntologyClass ioc, UtilityInformation ui)

Palauttaa utiliteettifunktion arvon parametrina saadusta InterestingOntologyClass-oliosta.

4.6 Luokka InterestingLink

Muuttujat

Luokalla on neljä muuttujaa.

private int utility

InterestingLink-olion utiliteetti.

private Image image

Mielenkiintoiseen linkkiin liittyvä kuva.

private Vector properties

InterestingLink-olion ominaisuudet.

private String linkname

InterestingLink-olion nimi.

Konstruktorit

Tällä luokallla on 8 konstruktoria.

public InterestingImage(String name)

Luo InterestingLink-olion nimenä saatu parametri.

public InterestingLink(Image im)

Luo InterestingLink-olion kuvanaan saatu parametri.

public InterestingLink(int util)

Luo InterestingLink-olion utiliteettinaan saatu parametri.

public InterestingLink(String name,Image im, int util)

Luo InterestingLink-olion nimenään, kuvanaan ja utiliteettinaan saadut parametrit.

public InterestingLink(String name, Image im)

Luo InterestingLink-olion kuvanaan ja nimenään saadut parametrit.

public InterestingLink(String name, int util)

Luo InterestingLink-olion nimenään ja utiliteettinaan saadut parametri.

public InterestingLink(Image im, int util)

Luo InterestingLink-olion kuvanaan ja utiliteettinaan saadut parametrit.

public InterestingLink(ResourceProperty ip,Image im)

Luo InterestingLink-olion kuvanaan ja (yhtenä) ominaisuutenaan saadut parametrit.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public Image getImage()

Palauttaa linkkiin liittyvän kuvan.

public boolean setImage(Image im)

Asettaa InterestingLink-olion kuvaksi saadun parametrin.

public Vector getProperties()

Palauttaa kuvaan liittyvät ominaisuudet.

public boolean addProprty(ResourceProperty ip)

Lisää linkkiin ominaisuuden.

public boolean setProperties(Vector props)

Asettaa saadun Vector-olion kuvan ominaisuuksiksi.

public boolean setLinkName(String name)

Asettaa linkin nimeksi saadun parametrin.

public boolean setUtility(int utility)

Asettaa linkin utiliteetiksi saadun parametrin.

public boolean getUtility()

Palauttaa linkin utiliteetin.

public int calculateUtility(UtilityFunction uf, UtilityInformation ui)

Laskee linkin utiliteetin UtilityInformation-olion pohjalta ja asettaa sen InterestingLink-olion utility-muuttujaksi. Tämän utiliteetin metodi myös palauttaa.

public int compareTo(Object o)

Palauttaa kutsujan ja parametrina saadun olion utiliteettien erotuksen.

public boolean equals(Object o)

Testaa kutsujan ja parametrina saadun olion samuutta. Jos kutsujan ja parametrin kuvat (image-muuttujat) ovat samat, niin palautetaan true, muuten false.

4.7 Luokka InterestingImage

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private int utility

InterestingImage-olion utiliteetti.

Konstruktorit

Tällä luokallla on 1 konstruktori.

public InterestingImage(Image image)

Luo InterestingImage-olion saadusta Image-olion.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public boolean setUtility(int util)

Asettaa kutsujan utiliteetiksi saadun kokonaisluvun.

public getUtility()

Palauttaa InterestingImage-olion utiliteetin.

public calculateUtility(UtilityFunction uf, UtilityInformation ui)

Palauttaa utiliteettifunktion arvon parametrina saadusta InterestingImage-oliosta.

public int compareTo(Object o)

Palauttaa kutsujan ja parametrina saadun olion utiliteettien erotuksen.

4.8 Luokka InterestingImageAccessCountUtilityFunction

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private MuseumModel model

Utiliteettifunktion tarvitsema MuseumModel-olio, katso getUtility().

Konstruktorit

Tällä luokalla on yksi konstruktori.

public InterestingImageAccessCountFunction(Image image)

Luo InterestingImageAccessCountUtilityFunction-olion saadusta Image-oliosta.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public boolean setUtility(int util)

Asettaa kutsujan utiliteetiksi saadun kokonaisluvun.

public int getUtility()

Palauttaa InterestingImage-olion utiliteetin suhteessa MuseumModel-olion suosituimpaan kuvaan.

4.9 Luokka InterestingImageAccessCountUtilityFunction

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private MuseumModel model

Utiliteettifunktion tarvitsema MuseumModel-olio, katso getUtility().

Konstruktorit

Tällä luokalla on yksi konstruktori.

public InterestingImageAccessCountFunction(Image image)

Luo InterestingImageAccessCountUtilityFunction-olion saadusta Image-oliosta.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public boolean setUtility(int util)

Asettaa kutsujan utiliteetiksi saadun kokonaisluvun.

public int getUtility()

Palauttaa InterestingImage-olion utiliteetin suhteessa MuseumModel-olion suosituimpaan kuvaan.

4.10 Luokka InterestingLinkFacetUtilityFunction

Huom! Tämä luokka sisältää sisäluokan MyFacetCombination, joka käsitellään seuraavaksi.

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private MuseumModel model

Utiliteettifunktion tarvitsema MuseumModel-olio, katso getUtility().

Konstruktorit

Tällä luokalla on yksi konstruktori.

public InterestingLinkFacetUtilityFunction(MuseumModel model)

Luo InterestingLinkFacetUtilityFunction-olion model-muuttujanaan saatu MuseumModel-olio .

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public boolean equals(Object o)

Testaa kutsujan fasettijuurien ja parametrina saadun fasettijuurien samuutta.

public int getUtility()

Palauttaa InterestingLink-olion utiliteetin suhteessa nykyisiin fasettivalintoihin. Tämä on toteutettu vertaamalla kuvan ontologiaripustuksia HiBrowse-valintoihin. Jokaista HiBrowse-valintaan sisältyvää fasettipuuta/solmua kohti tutkitaan, onko kuvassa samaan fasettipuuhun liittyvää solmua. Jos on, lasketaan solmujen välinen polku siirtymällä aina alemmalla tasolla olevasta solmusta ensimmäiseen vanhempaan, muuten lasketaan polun pituus HiBrowse-valinnan solmusta puun juureen. Utiliteetti muodostuu jakamalla maksimiutiliteetti näiden polkujen pituuksien summalla.

4.11 Luokka MyFacetCombination

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

OntologyClass facetRoot

OntologyClass facetNode

Konstruktorit

public MyFacetCombination(OntologyClass facetNode)

Luo MyFacetCombination-olion facetNode-muuttujanaan parametrina saatu OntologyClass-olio. Konstruktori hakee tämän OntologyClass-olion ensimmäisen esi-isän, jonka se sijoittaa MyFacetCombination-olion facetRoot-muuttujaksi.

Metodit

Tällä luokalla ei ole muita metodeja.

4.12 Luokka InterestingLinkGlobalAccessCountUtilityFunction

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private MuseumModel model

Utiliteettifunktion tarvitsema MuseumModel-olio, katso getUtility().

Konstruktorit

public InterestingLinkGlobalAccessUtilityFunction(MuseumModel model)

Luo InterestingLinkGlobalAccessCountUtilityFunction-olion saadusta Image-oliosta.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public getUtility(InterestingLink il, Profile profile)

Palauttaa InterestingLink-olion utiliteetin suhteessa MuseumModel-olion suosituimpaan kuvaominaisuuteen .

4.13 Luokka InterestingLinkImageLinkAccessCountUtilityFunction

Muuttujat

Tällä luokalla ei ole muuttujia.

Konstruktorit

Tällä luokalla ei ole kostruktoreja.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public getUtility(InterestingLink il, UtilityInformation ui)

Palauttaa InterestingLink-olion utiliteetin suhteessa malln suosituimman kuvaominaisuuteen. Tämän se tekee laskemalla InterestingLink-olion suosituimman ominaisuuden painon jaettuna olion lähdekuvan (sourceimage) suosituimman ominaisuuden painolla.

4.14 Luokka InterestingLinkTargetImageGlobalAccessCountUtilityFunction

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private MuseumModel model

Utiliteettifunktion tarvitsema MuseumModel-olio, katso getUtility().

Konstruktorit

public InterestingLinkTargetImageGlobalAccessUtilityFunction(MuseumModel model)

Luo InterestingLinkTargetImageGlobalAccessCountUtilityFunction-olion saadusta Image-oliosta.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public getUtility(InterestingLink il, Profile profile)

Palauttaa InterestingLink-olion kuvan utiliteetin suhteessa MuseumModel-olion suosituimpaankuvaan.

4.15 Luokka InterestingLinkTargetImageLocalAccessCountUtilityFunction

Muuttujat

Tällä luokalla ei ole muuttujia.

Konstruktorit

Tällä luokalla ei ole konstruktoreja.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public getUtility(InterestingLink il, UtilityInformation ui)

Palauttaa InterestingLink-olion kuvan utiliteetin suhteessa UtilityInformation-olion sisältämien PropertyValue-olioiden suosituimpaan kuvaan.

4.16 Luokka InterestingOntologyClass

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private int utility

Tämän olion utiliteetti.

Konstruktorit

public InterestingOntologyClass(OntologyClass oc)

Luo InterestingOntologyClass-olion saadusta OntologyClass-oliosta.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public int getUtility()

Palauttaa olion utiliteetin.

public boolean setUtility(int util)

Asettaa olion utiliteetiksi saadun util-muuttujan arvon.

public int calculteUtility()

Palauttaa olion utiliteetin.

public int calculateUtility

Asettaa utiliteetiksi maksimiutiliteetin (katso MAXUTILITY luokasta UtilityFunction) palauttaen sen.

public int compareTo(Object o)

Palauttaa kahden InterestingOntologyClass-olion, kutsujan ja muunnetun (eng. cast), utiliteettien erotuksen.

4.17 Luokka InterestingOntologyClassAccessCountUtilityFunction

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private MuseumModel model

Utiliteettifunktion tarvitsema MuseumModel-olio, katso getUtility().

Konstruktorit

public InterestingOntologyClassAccessUtilityFunction(MuseumModel model)

Luo InterestingOntologyClassAccessCountUtilityFunction-olion saadusta Image-oliosta.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public getUtility(InterestingLink ioc, UtilityInformation ui)

Palauttaa InterestingOntologyClass-olion utiliteetin suhteessa mallin suosituimpaan fasettiin.

4.18 Luokka PreviousImageUtilityFunction

Muuttujat

private MuseumModel model

Utiliteettifunktion tarvitsema MuseumModel-olio, katso getUtility().

Konstruktorit

Tällä luokalla ei ole konstruktoreja.

public PreviousImageUtilityFunction(MuseumModel model)

Luo PreviousImageUtilityFunction-olion saadusta Image-oliosta.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public getUtility(InterestingLink il, Profile profile)

Palauttaa InterestingLink-olion kuvan utiliteetin suhteessa Profiilissa aikaisemmin olleisiin kuviin. Mitä kauempana InterestingLink-olion kuva on aikaisempiin kuviin, sitä suurempi utiliteetti tälle kuvalle tulee. Aikaisemmat kuvat haetaan MuseumModel-oliosta.

public getUtility(InterestingImage, UtilityInformation ui

Saman kuin edellä, mutta utiliteetti lasketaan InterestingImage-oliolle.

4.19 Luokka RandomthresholdUtilityFunction

Muuttujat

Tällä luokalla ei ole muuttujia.

Konstruktorit

Tällä luokalla ei ole konstruktoreja.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public int getUtility(InterestingLink il, UtilityInformation ui)

Palauttaa InterestingLink-olion utiliteetin, joka on satunnaisluku. Jos satunnaisluku on suurempi 0.999, niin palautetaan nolla.

public int getUtility(InterestingImage ii, UtilityInformation ui)

Sama kuin edellä, mutta utiliteetti lasketaan InterestingImage-oliolle.

public int getUtility(InterestingOntologyClass ioc, UtilityInformation ui)

Sama kuin edellä, mutta utiliteetti lasketaan InterestingOntologyClass-oliolle.

4.20 Luokka RandomUtililityFunction

Muuttujat

Tällä luokalla ei ole muuttujia.

Konstruktorit

Tällä luokalla ei konstruktoreja.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public int getUtility(InterestingLink il, UtilityInformation ui)

Palautta satunnaisluvun InterestingLink-olion utiliteettina.

public int getUtility(InterestingImage ii,UtilityInformation ui)

Palautta satunnaisluvun InterestingImage-olion utiliteettina.

public int getUtility(InterestingOntologyClass ioc,UtilityInformation ui)

Palautta satunnaisluvun InterestingOntologyClass-olion utiliteettina.

4.21 Luokka UtilityFunctionChain

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private Vector weighedUtilityFunctionChain

Konstruktorit

Tällä luokalla ei ole konstruktoreja.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public boolean addUtilityFunction(WeighedUtilityFunction wuf)

Lisää utiliteettifunktion kutsujan weighedUtilityFunctionChain-olioon.

public boolean addUtilityFunction(UtilityFunction uf, int weight)

Lisää utiliteettifunktion kutsujan weighedUtilityFunctionChain-olioon parametrina saadulla painolla.

public Vector getUtilityFunctionChain()

Palauttaa kutsujan weighedUtilityFunctionChain-olion.

public boolean removeUtilityFunction(WeighedUtilityFunction uf)

Poistaa utiliteettifunktion weighedUtilityFunction-oliosta.

public boolean removeUtilityFunction(int index)

Sama kuin edellä, mutta utilitettifunktio poistetaan weighedUtilityFunction-oliosta indeksin määrämästä kohdasta.

public getUtility(InterestingLink il, UtilityInformation ui)

Palautta InterestingLink-olion utiliteetin jakamalla kaikkien ketjussa olevan utiliteettifunktioiden utiliteettien summan näiden utilieettifunktioiden painojen summalla.

public getUtility(InterestingImage ii,UtilityInformation ui)

Sama kuin edellä, mutta utiliteetti lasketaan InterestingImage-oliolle.

public getUtility(InterestingOntologyClass ioc,UtilityInformation ui)

Sama kuin edellä, mutta utiliteetti lasketaan InterestingOntologyClass-oliolle.

4.22 Luokka UtilityInformation

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private Profile profile

private Collection itemList

private Image sourceImage

Konstruktorit

Tällä luokalla on kaksi konstruktoria.

UtilityInformation(Profile profile, Collection itemList, Image sourceImage)

Luo UtilityInformation-olion saaduista Profile-,Collection- ja Image-olioista.

UtilityInformation(Profile profile, Collection itemList)

Luo UtilityInformation-olion saaduista Profile- ja Collection-olioista.

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public Image getSourceImage()

Palauttaa kutsuvan olion sourceImage-muuttujan.

public Image setSourceImage(Image si)

Asettaa saadun Image-olion kutsujan sourceImage-muuttujaksi.

public Profile getProfile()

Palauttaa kutsuvan olion profile-muuttujan.

public boolean setProfile(Profile p)

Asettaa saadun Profile-olion kutsujan profile-muuttujaksi.

public Collection getItemList()

Palauttaa kutsujan itemList-olion.

public boolean setItemList(Collection v)

Asettaa kutsujan itemList-muuttujaksi saadun Collection-olion.

4.23 Luokka WeighedUtililityFunction

Muuttujat

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

private UtilityFunction uf;

private int weight

Konstruktorit

Tällä luokalla on yksi konstruktori

WeighedUtilityFunction(UtilityFunction uf, int weight)

Luo WeighedUtilityFunction-olion saaduista UtilityFunction-oliosta ja sen painosta (weight).

Metodit

Tähän mahdollisesti lause tai kaksi kuvaavaa höpinää.

public int getWeight()

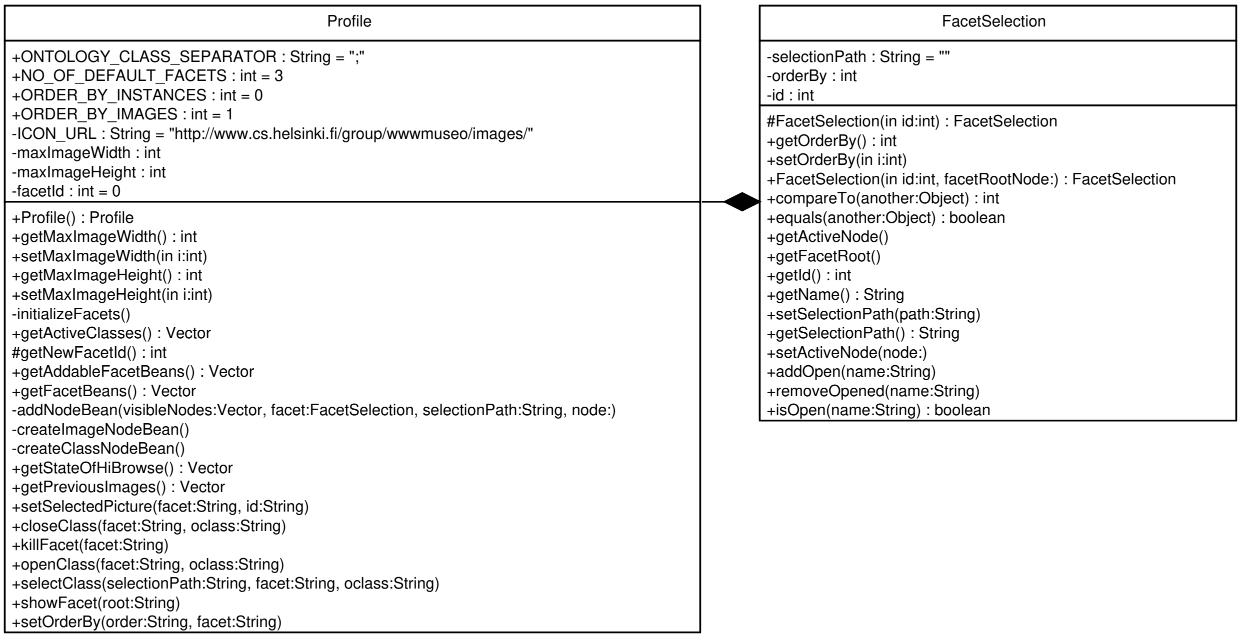
Palauttaa kutsujan weight-muuttujan arvon.

public boolean setWeight(int weight)

Asettaja kutsujan weight-muuttujaksi parametrina saadun luvun.

public UtilityFunction getUtilityFunction()

Palauttaa kutsujan uf-muuttujan.



Kuva 5: Profiilin luokkakaavio

public boolean setUtilityFunction(UtilityFunction uf)

Asettaa kutsujan uf-muuttujaksi saadun UtilityFunction-olion.

5 Profiilimoduuli

Profiilimoduulin tehtävä on varastoida käyttäjäkohtaista tietoa.

5.1 Yleiskuvaus

5.2 Moduulin ulkoinen rajapinta

Suodattimelle tarjottavat palvelut ovat HiBrowsen tila ja edelliset kuvavalinat. Näiden lisäksi profiili tarjoaa käyttöliittymälle palvelun, jonka auttaa käyttöliittymää näyttämään hiBrowsen.

Käyttöliittymä on vastuussa siitä, että Profile-luokkaan taltioitu HiBrowsen tila sekä valitut kuvat ovat aina ajan tasalla.

5.3 Moduulin sisäinen rajapinta

Profile-luokalla on apunaan FacetSelection-luokka, joka varastoi tiedot yhden käyttäjän yhden fasetin tiedot.

5.4 Luokka Profile

Modulin pääluokka. Tämän luokan ilmentymän käyttöliittymä varastoi sessioon jokaiselle käyttäjällä.

Muuttujat

public static final String ONTOLOGY_CLASS_SEPARATOR

Käytetään jaettaessa facetin valitun luokan täsmentävää merkkijonoa osiin.

public static final int NO_OF_DEFAULT_FACETS

Kertoo kuinka monta facettia käyttäjälle näytetään alkutilanteessa.

public static final int ORDER_BY_INSTANCES

Kun fasetti halutaan järjestää ilmentmien mukaan kutsutaan FacetSelection.setOrdrBy(Profile.ORDER_BY_INSTANCES).

public static final int ORDER_BY_IMAGES

Kun fasetti halutaan järjestää kuvien mukaan kutsutaan FacetSelection.setOrdrBy(Profile.ORDER_BY_IMAGES).

private static final String ICON_URL

Ikoni kuvien perus URL.

private int maxImageWidth/maxImageHeight

Kertovat käyttöliittymässä mäytettävien kuvien maksimi leveyden ja korkeuden.

private int facetId

Käytetään jaettaessa ykilöllisiä tynniste numeroita faseteille.

private Hashtable intersect

Käytetään apuna muodostettaessa leikkausta.

private Vector selectedFacets

Pitää sisällään FacetSelection-ilmenntymiä, eli kaikki ne facetiti, jotka näkyvät ruudulla.

private Vector previousSelections

String-olioita, joissa kaikkien katseltujen kuvien tunnisteet.

private Vector allFacetRoots

private Filter filter

Ilmentymä, joka tarjoaa Filter-luokan työkalut profiilin käyttöön.

private MuseumModel model

Ilmentymä, joka tarjoaa MuseumModel-luokan työkalut profiilin käyttöön.

Konstruktorit

public Profile()

Luo uuden Profile-olion ja alustaa oletus fasetit.

Metodit

public int getMaxImageWidth()/getMaxImageHeight()

Palauttaa maksimi kuvan koon.

public int setMaxImageWidth(int i)/setMaxImageHeight(int i)

Asettaa maksimi kuvan koon.

public Vector getActiveClasses()

Palauttaa Vector-olion, jossa OntologyClass-ilmentyiä. Tämä metodi palauttaa kaikki käyttäjän tällähetkellä aktiiviset luokat.

public Vector getAddableFacetBeans

Palauttaa Vector-olion, jossa AddableFacetBean-ilmentymiä. Tämä metodi palauttaa tarvittat tiedot käyttöliittymälle, jotta lisättävien fasettien lista voidaan muodostaa.

public Vector getFacetBeans()

Palauttaa Vector-olio, jossa FacetBean-ilmentymiä. Tämä metodi palauttaa tarvittavat tiedot käyttöliittymälle, jotta valitut fasetit voiddan näyttää käyttäjälle.

public Vector getStateOfHiBrowse()

Palauttaa Vector-olion, jossa kaikki näkyvillä olevia fasetteja vastaavat FacetSelection-ilmentymät.

public Vector getPreviousImages()

Palautaa Vertor-olion, jossa kaikki käyttäjän kuvavalinnat String-olioina.

public void setSelectedPicture(String facet, String oclass)

Lisää kuvan käyttäjän kuvavalintojen listaan. Tätä metodia voidaan parantaa niin, että se lisäisi fasetin painoa, josta kuvavalinta tehtiin.

public void closeClass(String facet, String oclass)

Sulkee fasetin luokan.

public void openClass(String facet, String oclass)

Avaa fasetin luokan.

public void selectClass(String selectionPath, String facet, String oclass)

Asettaa fasetin luokan aktiiviseksi.

public void showFacet(String root)

Lisää fasetin näkyvien fasettien listaan.

public void setOrderBy(String order, String facet)

Asettaa fasetin järjestämistavan. Muuttujan order on oltava "instances", jos järjestys halutaan olevan ilmentymien mukaan. Muuta arvot asettavat järjestyksen kuvien mukaan.

private static ClassNodeBean createClassNodeBean(String id, int depth, String targetURL, String URLParameters, String iconURL, int amountOfImages, int amountOfAllImages, int amountOfInstances, String expanderOrCollapserHTML, String header

Apumetodi ClassNodeBean-olion luomiseksi.

private static ImageNodeBean createImageNodeBean(String id, int depth, String targetURL, String URLParameters, String iconURL,String instanceNames)

Apumetodi ImageNodeBean-olion luomiseksi.

private void addNodeBean(Vector visibleNodes, FacetSelection facet, String selectionPath, OntologyClass node, boolean onlyContent, int depth, boolean rootOpen)

Apumetodi, jota käytetään facetNodeBean-olion sisältämien NodeBean-olioiden luontiin. Tätä metodia kutsutaan rekursiiviseti.

private void initializeFacets()

Konstruktori kutuu tätä metodia alustamaan fasettien oletusarvot.

5.5 Luokka FacetSelection

Profile-luokan apuluokka yhden fasetin tilan säilömiseen.

Muuttujat

private OntologyClass activeNode

Kertoo tämän fasetin aktiiviseksi valitun luokan.

private OntologyClass facetRoot

Tämän fasetin juuren luokka.

private String selectionPath

Tähän muuttujaan on tallennettu polku valittuun luokkaan juuresta. Merkkijonossa on alussa juuren luokkan tunniste ja siitä eteenpäin muiden luokkine tunnisteita kunnes päästään valitutun luokkaa. Luokkine tunnisteet on eroiteltu käyttäen Profiili luokassa määriteltyä vakiota ONTOLOGY_CLASS_SEPARATOR.

private int id

Tämän fasetin yksilöllinen tunniste numero.

private Vector openedNodes

Vector-olio, johon tallennettu kaikkien luokkien tunnisteet, jotka käyttäjä on avannut.

Konstruktorit

public FacetSelection(int id)

Luo uuden FasetSelection-olion tunnisteella id.

public FacetSelection(int id, OntologyClass facetRotNode)

Luo uuden FasetSelection-olion tunnisteella id ja juureksi asetetaan facetRootNode.

Metodit

public void setOrdrBy(int i)

Asettaa järjestystava.

public int getOrdrBy()

Palauttaa järjestystavan.

public int compareTo(Object another)

Vertailee kahta FacetSelection oliota.

public boolean equals(Object another)

Tarkistaa kahden FacetSelection olion samuuden.

public OntologyClass getActiveNode()

Palauttaa tämän fasetin valitun luokan.

public OntologyClass getFacetRoot()

Palauttaa tämän fasetin juuren.

public int getId()

Palauttaa tämän fasetin yksilöllisen tunnisteen.

public String getName()

Palauttaa tämän fasetin nimen.

public void setSelectionPath(String path)

Asettaa tämän fasetin valintapolun.

public String getSelectionPath()

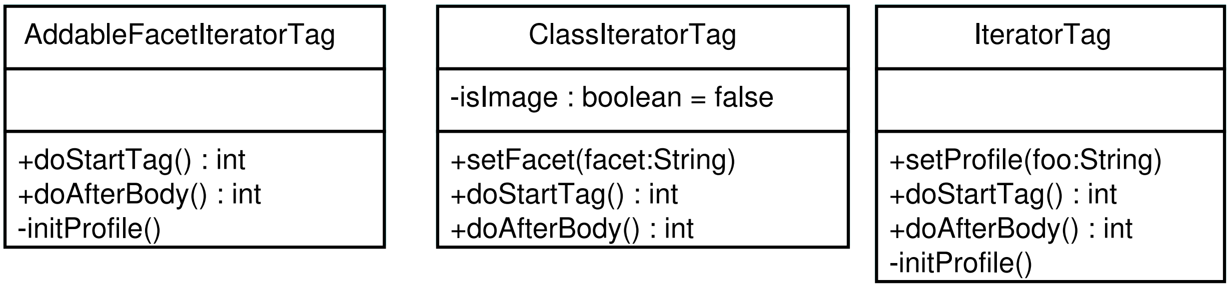
Palauttaa tämän fasetin valintapolun.

public void setActiveNode(OntologyClass node)

Asettaa tämän fasetin valitun luokan.

public void addOpen(String name)

Asettaa tähän fasettiin avatuksi “name”-nimisen luokan.



Kuva 6: HiBrowse tag kirjasto

public void removeOpened(String name)

Poistaa avoinna olevien luokkien listasta “name”-nimisen luokan.

public boolean isOpen(String name)

Onko “name”-niminen luokka avoinna?

6 Käyttöliittymämoduuli

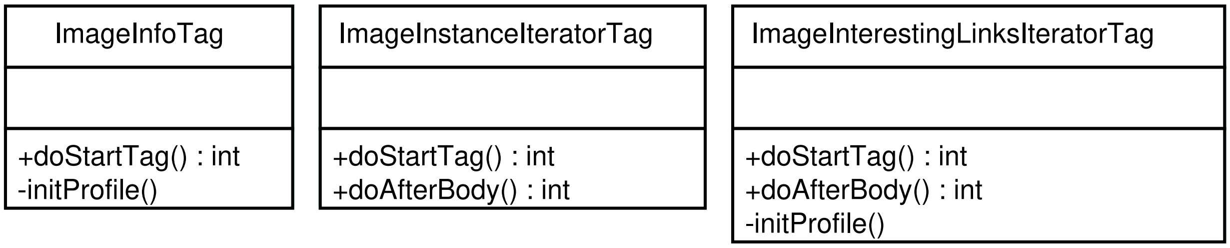
Käyttöliittymämodulin tehtävä on muodostaa käyttäjälle näytettävä näkymä HiBrowsesta, kuvista ja mielenkiintoisista linkeistä.

6.1 Yleiskuvaus

Käyttäliittymä on toteutettu käyttäen JSP custom tag library tekniikkaa. Käyttöliittymä jakaantuu kahteen osaan HiBrowse-näkymään(hiBrowse.jsp) ja kuvasivuun(result.jsp), joka sisältä myös mielenkiintoiset linkit.

6.2 Moduulin ulkoinen rajapinta

Käyttöliittymä käyttää proofiilin tarjoamia palveluita näkymän muodostamiseen. Tärkein luokka on Profile. Sitä käytetään luomaan pavut, jotka sisältävät kaiken tarvittavan tiedon.



Kuva 7: Result tag kirjasto

Käyttöliittymän tehtäväksi jä yksinkertaisesti papujen läpikäynti ja niiden tietojen näyttäminen halutulla tavalla.

6.3 HiBrowse tag-kirjaston kuvaus

HiBrowse tag-kirjastoa käytetään HiBrowse-näkymän muodostamseen.

6.3.1 iterateFacets

IteratorTag-luokka toteuttaa tämän tagin. Tämä tag asettaa pageContext-olioon "facet"-nimellä FacetBean-olion, josta voidaan tulostaa tietoja JSP-sivulla. Tämä tag käy läpi kaikki FacetBean-oli, jotka Profile.getFacetBeans()-metodi palauttaa. Tagille on annettava attribuutti "profile", mutta sen arvolla ei ole tässä versiossa merkitystä.

6.3.2 iterateNodes

ClassIteratorTag-luokka toteuttaa tämän tagin. Tämä tag käy läpi kaikki fasetin näkyvillä olevat luokat ja kuvat. Mikäli tällä kierroksella käsiteltävänä oleva fasetin solmu on luokka asetetaan pageContext-olioon "isImage"-nimellä merkkijono "false" ja "classNode"-nimellä ClassNodeBean-ilmentymä, jota käytetään tämän luokan näyttämiseen näytöllä. Jos taas on kyseessä kuva, asetetaan pageContext-muuttujaan "isImage"-nimellä "true"-merkkijono ja "imageNode"-nimellä ImageNodeBean-ilmentymä. Tagille on annettava attribuutto "faacet", mutta sen arvolla ei ole tässä veriossa merkitystä.

6.3.3 iterateAddableFacets

AddableFacetIteratorTag-luokka toteuttaa tämän tagin. Tämä tag läpikäy profiililta saamansa listasta faseteista, jotka voidaan lisätä näkymää ja luo käyttäjällä sopivan käyttöliittymäkomponentin. pageContext-oloioon asetetaan joka kierroksella nimelä “addableFacet” AddableFacetBean-olio. Tämän pitäisi riittää, mutta testattaessa jostain syystä tämä ei toiminut. Tagin toimintaan saattamiseksi pageContext-muuttujaan asetetaan AddableFacetBean-olion sisältämät tiedot merkkijonoina.

6.4 Result tag-kirjaston kuvaus

Result tag-kirjastoa käytetään kuvasivun eli tulosnäkymän muodostamiseen.

6.4.1 imageInfo

ImageInfoTag-luokka toteuttaa tämän tagin. pageContext-olioon asetetaan seuraavat merkkijonot: “imageDescription”, “imageHeight”, “imageWidth”, “mageURL” ja “imageThumbURL”.

6.4.2 imageInstanceIterator

ImageInstanceIteratorTag-luokka toteuttaa tämän tagin. Tämä tagi käy läpi kuvassa esiintyvät ilmentymät, ja asettaa kunkin kohdalla pageContext-olioon nimillä “label” ja “value” merkkijonot, jotka kuvaavat ilmentymän tietoja.

6.4.3 imageInterestingLinksIterator

ImageInterestingLinksIteratorTag-luokka toteuttaa tämän tagin. PageContext-olioon asetetaan kullakin kierroksella linkin tiedot merkkijonoina. “thumbURL”, “linkName”, “linkImageId”, “linkUtility” ja “starsHTML”.

6.5 HiBrowse JSP-sivun kuvaus

hiBrowse.jsp koostuu kolmesta iteraatio tagista. iterateFacests-tagia käytetään kaikkien näkyvillä olevien fasettien läpikäyntiin. Tämän silmukan sisällä on iterateNodessilmukka, joka käy läpi kaikki näkyvät luokat ja kuvat. Näiden sisäkkäisten silmukoiden jälkeen on näistä täysin erillinen osa, jossa muodostetaan pudotuslista lisättävistä faseteista iterateAddableFacets-tagilla.

6.6 Result JSP-sivun kuvaus

result.jsp koostuu kolmesta osasta, joista ensimmäisessä näytetään itse kuva ja sen kuvaus. Tämänjälkeen käydään läpi kuvan ilmentymät. Viimeinen vaihe on mielenkiintoisten linkkien näyttäminen.

7 Projektissa toteutetut prototyypit (SS)

Tähän muutaman lauseen alkuhöpinä.

7.1 Prototyyppien esittely

Todetaan, että tehtiin kaksi protoa. Sanotaan milloin ensimmäinen hylättiin ja toinen aloitettiin.

7.2 Prototyyppien vertailua

Mitä eroa protoilla on? Miksi ensimmäinen hylättiin? MItä hyvää/huonoa protoissa on toisiinsa verrattuna. Onko eka proto pelkästään huono ilman mitään hyvää? Hyvät perustelut, koska tehty ratkaisu oli varsin radikaali.

8 Testaus

Tähän muutaman lauseen alkuhöpinä.

8.1 Järjestelmän yleistestaus

Miten testattiin, että kaikki moduulit toimivat yhdessä? Miten järjestelmää voidaan testata tulevaisuudessa?

8.2 Metatietomoduulin testaus

Moduuli testattiin testiluokalla TestMuseumModel2. Luokassa olevat testit vastaavat luvussa esitettyjä palveluja. Epäsuoraa testausta saatiin myös muitten moduulien testausten yhteydessä. Testaus suoritettiin vain yhdellä ontologialla, joten sitä ei voida kehua erityisen kattavaksi.

8.3 Suodattimen testaus (EM)

Suodattimen toiminta testattiin erillisellä linkkien generoinnin oikeellisuuden testaamisella sekä jatkuvalla koko järjestelmän käyttötestauksen aikaisella utiliteettifunktioiden arvojen tarkkailulla.

Linkkien generointi testattiin FilterTest -luokalla, vertaamalla valituista esimerkkikuvista muodostettuja linkkejä käsin etsittyihin.

Utiliteettifunktioiden toimivuus varmistettiin useamman viikon kestäneellä koko järjestelmän käytön aikaisella tarkkailulla, jossa ohjelman lokiin kirjattiin kaikki poikkeamat utiliteetin mahdollisista arvoista (0-65535). Kaikki löydetyt virheet tutkittiin ja korjattiin.

Varsinaista utiliteettifunktioiden oikeaa toimintaa ei siis kuitenkaan tarkistettu, ainoastaan, että funktioiden arvot vaihtelevat eri linkkien välillä ja että vaihtelu jää utiliteettifunktioiden mahdollisten arvorajojen sisälle.

8.4 Profiilimoduulin testaus

Profiilimodulin luokkien testaus toteutettiin samalla käyttöliittymämoduulin testauksen kanssa.

8.5 Käyttöliittymämoduulin testaus

Minkäänlaista testaussuunnitelmaa ei tehty, eli varsinaista testausta ei toteutettu.Testaaminen jäi käytännössä eri moduulien ja kokojärjestelmän kokeilemiseksi.Jatkokehityksessä olisi syytä testata ensisijaisesti virhetilanteet ja niistä toipuminen.

8.6 Testauksen ongelmia

Miksi emme testanneet kunnolla? Mitä ei edes voinut testata kunnolla?

9 Toteutuksen arviointia (KJ, SS, EM, TR, RH, PL)

Tässä luvussa kartoitetaan järjestelmän ongelmat ja arvoidaan osajärjestelmien toimivuutta ja kehitysmahdollisuuksia.

9.1 Toteutuksen onnistuminen ja puutteet

Projekti on kokonaisuudessaan onnistunut. Toteutettu järjestelmä onnistuu näyttämään periaatteen toimivuuden ja vaikka ominaisuuksissa on puutteita eikä suoritettu käytettävyystestejä, niin toteutetuissa ominaisuuksissa ei ole tunnettuja toiminnallisuuspuutteita.

9.2 Metatietomoduulin toteutus ja sen arvointi

Metatietomoduuli kirjoitettiin täysin uudelleen ensimmäisen prototyypin valmistuttua. Suurin osa ongelmista oli jo ratkaistu, mutta moduulin rakenne jätti toivomisen varaa. Tämän lisäksi toiseen versioon lisättiin ominaisuuksia, varauduttuun paremmin tuleviin ominaisuuksiin ja optimoitiin joitain usein käytettyjä toimintoja hajautustaulukkoja käyttäen. Seuraavaksi on esitelty joitain parannusehdotuksia seuraaville versioille.

9.2.1 Linkit

Parannuksen varaa olisi ainakin linkkien käsittelyssä; tällä hetkellä linkki on suodatinmoduulin tulkinta kahden kuvan yhteenkuuluvuudesta yhden tai useamman ominaisuuden perusteella. Mikäli kahden ontologiaresurssien välisestä linkistä tehtäisiin luokka, jonka ominaisuuksina ovat linkin muodostavat ominaisuudet, voidaan helposti tehdä linkkiketjuja, jotka vastaavat useamman nykyjärjestelmän klikkausten yhdistäminen yhdeksi. Suodatinmoduulin tehtävänä olisi sitten päätellä, miten näitä käyttämällä saataisiin aikaan mielenkiintoempia linkkejä.

9.2.2 Item-luokat

Yksi parannusehdotus olisi muokata Item-luokkarakennetta niin, että erillisiä Image-, Movie-, Book- ym. luokkia ei tarvittaisi, vaan AbstactItem ja ConcreteItem pystyisivät luontevasti kuvaamaan minkä tahansa resurssin, jolloin järjestelmä ei olisi sitoutunut vain tietyntyyppisiin näyttelyihin.

9.2.3 Kuvattavat resurssit

Kuvassa (tai kirjassa, elokuvassa ym.) esiintyviä "tosimaaliman resursseja", kuvaava AbstactItem-luokan toteutus ei ole valmis. Instansseja pitäisi muokata niin että ne aina osoittavat yhteen AbstactItem-olioon, jotta esimerkiksi sama henkilö eri rooleissa (Toimihenkilö, Yksittäishenkilö) eri kuvissa voidaan tunnistaa samaksi. Tämä vaatii nykyjärjestelyllä ontologian muuttamista. Parempi ratkaisu on varmaan olemassa.

9.2.4 Malli tietokannassa

Jena tukee mallin lukemista tietokannasta. Tämän ominaisuuden käyttö varmasti nopeuttaisi mallien käsittelyä ja vaikuttaa muutenkin järkevämmältä kuin puhtaiden rdf/s-tiedostojen lukeminen.

9.3 Suodatinmoduulin toteutus (EM)

Suodatinmoduulin alkutoteutus oli helppo. Utiliteettifunktioformalismi antoi mahdollisuuden toteuttaa ensin yksinkertainen pohja, jonka päälle utiliteettifunktioita voitiin rakentaa yksi kerrallaan. Tämä mahdollisti suodattimen saattamisen toimintaan ja käyttötestaukseen jo hyvin varhaisessa vaiheessa ohjelmointityötä. Utiliteettifunktiojoukon kasvaessa suodatin vain tarkentui.

Ohjelmiston uudelleensuunnittelun yhteydessä suodattimestakin alettiin koodata yleisempää versiota, joka lisäsi suodatinmalliin utiliteettifunktioiden ja utiliteettifunktioketjujen kanssa hyvin samanlaisen yleisen suodattimen ja suodatinketjun käsitteen, mutta tämä jätettiin ottamatta käyttöön liian suurina riskinä.

Parannusehdotuksia

Linkkien generointi

Tällä hetkellä linkkien kohteiksi valitaan ainoastaan kuvia, joilla on yhteisiä instansseja valitun kuvan kanssa. Linkkejä kannattaisi generoida muillakin perusteilla. Erityisesti pitäisi pystyä tunnistamaan samat entiteetit eri rooleissa (tosin se ei ole niinkään suodattimen ongelma). Myös pelkästään samoja ontologialuokkia sisältävät kuvat tulisi ehdottomasti lisätä linkattavien joukkoon.

Linkkien määrän kasvattamiseksi kannattaisi myös harkita esimerkiksi aikaisemmin läpikäytyjen kuvien linkkien ottamista mukaan (tähän voitaisiin sitten tietysti liittää linkin generoivan kuvan historiaetäisyyttä mittaava utiliteettifunktio)

Utiliteettifunktiot

Ohjelmassa on jo kunnioitettava määrä erilaisia utiliteettifunktioita, mutta niitä voi aina keksiä lisääkin, esimerkiksi:

- Kuinka hyvin kuva sopii käyttäjän aikaisempiin fasettivalintoihin
- Demografiatietoihin perustuvat utiliteettifunktiot
- Kuinka lähellä kuva on kuvia, joita käyttäjä on aikaisemmin katsonut

Utiliteettifunktioiden kertoimet

Utiliteetin painotetun keskiarvon laskemisessa käytetyt kertoimia ei ole viritetty. Käsin tehtävän virittämisen lisäksi voisi olla hyvin palkitsevaa kokeilla ainakin osittaista dynaamista kertointen muodostusta järjestelmässä. Käyttäjän profiiliin tallennettaisiin valittujen kuvien lisäksi seuratut linkit ja linkkiä kohden jokaisen eri utiliteettifunktion laskema utiliteetti.

Uusia linkkejä generoitaessa utiliteettifunktioiden painoja muutettaisiin sitten dynaamisesti siten, että niitä utiliteettifunktioita, joiden suositusta on aikaisemmin seurattu (jotka ovat siis antaneet korkean utiliteetin käyttäjän valitsemille linkeille) painotetaan enemmän kuin sellaisia, jotka ovat antaneet seuratuille linkeille huonoja utiliteettiarvoja.

Näin järjestelmä mukautuisi huomattavasti paremmin käyttäjään - jos järjestelmä huomaa, että käyttäjän mieltymykset eivät esimerkiksi ollenkaan vastaa keskivertokäyttäjää, huomioi järjestelmä vähemmän yleisten suosituimmuusfunktioden arvoja.

Tietysti myös koko järjestelmän oletuspainoarvoja voitaisiin päivittää näin dynaamisesti.

Toinen harkittava asia on, tarvitaanko kokonaisutiliteetin laskemiseen yleensä muita operaattoreita kuin painotettua keskiarvoa. Joidenkin utiliteettifunktioiden kohdalla esimerkiksi suora kertolasku tuntuisi paremmalta vaihtoehdolta. Esimerkiksi jos linkki on jo nähty, sen utiliteetin pitäisi olla pieni riippumatta siitä, kuinka kiinnostava se muuten on. Osaksi tämä hoituu painottamisellakin, edellisessä esimerkissä jo-nähty -utiliteettifunktion paino voidaan asettaa suuremmaksi kuin kaikkien muiden yhteensä. Tällainen ratkaisu ei kuitenkaan ole erityisen kaunis.

9.4 Profiilimoduulin toteutus

Profiilin kaikki toiminta on ehkäpä turhaakin keskitetty vain yhteen luokkaan.

Suunnitelman mukaan profiilimoduuli olisi pitänyt toteuttaa täysin erillään käyttöliittymästä. Ensimmäisessä protossa nämä kaksi moduulia kuitenkin sotkeentuivat jonkinverran. Toisessakin protossa profiili tuottaa edelleen vain käyttöliittymälle suunnattuja palveluita(papujen luonti). Tulevaisuudessa olisi fiksua erittää nämä kaksi moduulia poistamalla profiilimoduulista papujen luonti ja siirtä ne toiminnot käyttöliittymämoduuliin.

Tämän lisäksi proofilin algoritmeja voisi varmasti optimoida tehokkaimmiksi.

9.5 Käyttöliittymämoduulin toteutus

Ensimmäinen proto toteutettiin periaattella “kunhan se vaan toimii”. Toiseen protoon toteutettiin käyttöliittymän muodostamiseen tarkoitettuja uudelleen käytettäviä komponentteja, joden avulla käyttöliittymän uudelleen toteutus on jatkossa helppoa.

Liite 1. Järjestelmän asennusohje

Järjestelmän asennusohje (EM)

Tässä luvussa kuvataan järjestelmän asennus puhtaaseen koneeseen. Jos koneessa on jo tarpeeksi tuoreet versiot tarvittavista taustaohjelmista, voidaan siirtyä suoraan varsinaisen järjestelmän asennukseen ja konfigurointiin.

Vaadittavat apuohjelmat

Järjestelmä tarvitsee pohjakseen vähintään versiota 1.1 JSP-spesifikaatiosta tukevan servlet-ympäristön. Itse käytimme Tomcatin versiota 4.0.3 ilman Apachea, jolle konfiguraatio-ohjeemmekin kirjoitettu. Projektin kotisivuilta (http://jakarta.apache.org/tomcat/index.html) ladattava binääriversio toimii purkamisen jälkeen käytännössä suoraan.

RDF-metadatan käsittelyyn käytetään HP Labsin Jena-työkalupakettia, jonka lataaminen onnistuu osoitteesta http://www.hpl.hp.com/semweb/.

Järjestelmän tiedostot

Järjestelmä koostuu varsinaisen ohjelmiston luokkakirjastosta w3m.jar, konfiguraatiotiedostosta web.xml, käyttöliittymän jsp-, html- ja kuvatiedostoista, sekä varsinaisesta rdf- ja kuvadatasta.

Asennus

Asenna Tomcat omien ohjeidensa mukaan.

- Muokkaa Tomcatin conf/server.xml -tiedostoa, lisää <Server> -tagin sisään

```
<!-- wwwmuseo Context -->
<Context path="/wwwmuseo" docBase="wwwmuseo" debug="0">
  <Logger className="org.apache.catalina.logger.FileLogger"
          prefix="localhost_wwwmuseo_log." suffix=".txt"
          timestamp="true"/>
```

```
</Context>
```

- luo webapps/wwwmuseo hakemisto.
- luo webapps/wwwmuseo/WEB-INF hakemisto.
- luo webapps/wwwmuseo/WEB-INF/lib hakemisto.
- kopioi käyttöliittymän jsp-tiedostot webapps/wwwmuseo hakemistoon
- kopioi järjestelmän konfiguraatiotiedosto web.xml webapps/wwwmuseo/WEB-INF hakemistoon
- kopioi käytettävät museon kuvat jonnekin www:ssä näkyvään paikkaan (alihakemisto pictures/0 alkuperäisille kuville ja alihakemisto pictures/1 thumbnaileille)
- muokkaa lähdekoodista (W3m.java) url-parametri osoittamaan kuvien sijaintipaikkaa webissä
- kopioi järjestelmän RDF-tiedostot paikkaan, joka on ohjelmiston ulottuvissa (webapps/wwwmuseo/WEB-INF/data on oletuspaikka, jota käyttämällä koodia ei tarvitse muuttaa)
- etsi koodista RDF-tiedostoihin viittaavat polut ja muuta ne oikeiksi
- jos olet muuttanut tiedostopolkuja koodissa, käännä uusi versio w3m.jar-kirjastosta
- kopioi w3m.jar hakemistoon webapps/wwwmuseo/WEB-INF/lib
- käynnistä Tomcat sen omalla start-skriptillä
- järjestelmän pitäisi olla toiminnassa osoitteessa
  http://tomcat-palvelimennimi:tomcatinportti/wwwmuseo/

Liite 2. Lyhyitä määritelmiä

Seuraavassa taulukoituna lyhyitä määritelmiä käsitteistä ja työkaluista, joita olemme käyttäneet.

Termi:	Määrittely:
CSS	Cascading Style Sheets. Merkkauskieli, jolla voidaan määritellä esim. HTML dokumenttien ulkoasu. Tarkempaa tietoa osoitteessa www.w3.org/Style/CSS/.
DAML+OIL	DAML+OIL on erityisesty semanttista www:tä varten kehitetty version RDF-kielestä. Tarkempi kuvaus osoitteessa http://www.daml.org/2001/03/daml+oil-index.
Dublin Core	Dublin Core on metadatan standardisointiin pyrkivä projekti. Tarkempaa tietoa löytyy projektin kotisivuilla osoitteessa http://dublincore.org/.
Fasetti	Hakukäsite. Muodostuu saman aihealueen sisällä samaa aspektia kuvaavista asioista tai niitä tarkoittavista käsitteistä. Niiden käsitteiden joukko, joita voidaan pitää haussa vaihtoehtoisina. Fasetti on siis käsiteryväs.
HTML	Hypertext Markup Language. WWW-sivujen kuvaamiseen käytettävä kieli. Tarkempi kuvaus osoitteessa www.w3.org/TR/html4/.
J2EE	Java 2 Enterprise Edition. Sunin kehittämä laajennus java-ympäristöön. Tarkempaa tietoa osoitteessa http://java.sun.com/.
JRE	Java Runtime Environment. Sunin kehittämä ohjelmointiympäristö. Tarkempaa tietoa osoitteessa http://java.sun.com/.
JavaBeans	Teknologia, jolla voidaan luoda ohjelmakomponentteja Java-ympäristössä. Katso lisää osoitteesta http://java.sun.com/products/javabeans/docs/.

Jena	Javalla toteutettu työkalu, jolla voidaan käsitellä RDF-malleja. Tarkempi kuvaus osoitteessa www.hpl.hp.com/semweb.
JSP	JavaServer Pages. Käyttää XML:n kaltaisia tageja kapseloimaan logiikan, joka luo web-sivun. Tarkempaa tietoa osoitteessa http://java.sun.com/products/jsp/.
Mopsi	Alkuperäinen kuvatietokanta josta järjestelmän kuvat ja annotointidata on haettu.
Ontologia	Tiedon riippuvuussuhteiden esittäminen puumaisena rakenteena. Puun juurena on esim. talo, jonka lapsia ovat ylä- ja alakerta ja lapsenlapsia ullakko ja keittiö.
Protégé	Protégé 2000 on ontologiaeditori jolla järjestelmän ontologiat ja instassit on tehty. Projektin kotisivu osoitteessa http://protege.stanford.edu/.
RDF	Resource Description Framework. Kieli, jolla voidaan yhdistää erilaista XML-muodossa olevaa tietoa toisiinsa ontologioiden avulla. Tarkempaa tietoa osoitteessa www.w3.org/RDF/.
XML	Extensible markup language. Metakieli rakenteisten dokumenttien esittämiseen. Tarkempaa tietoa osoitteessa www.w3.org/TR/2000/REC-xml-20001006.