

Dogma

Suunnitteludokumentti

ohtu06-dogma-list@cs.helsinki.fi

Helsingin yliopisto, Tietojenkäsittelytieteen laitos

Ohjelmistotuotantoprojekti

20/12/2006

Kurssi

581260 Ohjelmistotuotantoprojekti (9 op, 6 ov)

Projektiryhmä

Bjorkgren Joakim
Kuronen Ville
Ristola Toni
Tani Antti
Vihavainen Arto

Asiakas

Niinivaara Olli

Johtoryhmä

Taina Juha
Moilanen Ilari

Kotisivu

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

Versiohistoria

<i>Versio</i>	<i>Päiväys</i>	<i>Muutokset</i>
0.1	20.09.2006	Ensimmäinen versio
0.5	06.11.2006	Korjattu versio
0.65	13.11.2006	Tarkastuksessa huomattujen puutteiden korjaus
0.7	20.12.2006	Useita korjauksia

Sisältö

1.Johdanto.....	1
1.1 Sanasto.....	1
1.2 Dokumentin rakenne.....	1
2.Arkkitehtuurin kuvaus.....	2
2.1 Arkkitehtuurikaavio.....	2
2.2 Päätoimintojen kuvaukset.....	2
2.2.1 Resurssiselaimen avaaminen	2
2.2.2 Resurssien hakeminen hakutaulukon avulla	2
2.2.3 Resurssien järjestäminen, hakeminen ja selaaminen	3
2.2.4 Resurssien valitseminen resurssitaulukossa	3
2.2.5 Lomakenäkymä ja resurssien muokkaus	3
2.2.6 Uuden resurssin luominen	4
2.2.7 Resurssien yhdistäminen, puolittaminen ja poistaminen	4
2.2.8 URI-taulukko ja toiminnot	4
2.2.9 Vapaamuotoinen SQL-kysely	4
2.2.10 Resurssien tarkastelu verkkonäkymässä.....	5
2.2.11 Viimeisimpien hakujen listaaminen.....	5
2.2.12 Tapahtumien tallentaminen loki-tiedostoon.....	5
2.2.13 Muut toiminnallisuudet lyhyesti.....	5
2.3 Pääkomponentit lyhyesti.....	5
2.3.1 Käyttöliittymä.....	5
2.3.2 Loki.....	5
2.3.3 Resurssikanta.....	6
2.3.4 Päivittäjä.....	6
2.3.5 Verkkonäkymä.....	6
3.Pääkomponenttien rajapinnat.....	6
3.1 Käyttöliittymä.....	6
3.2 Loki.....	6
3.3 ResourceBase.....	7
3.3.1 BaseConnection.....	7
3.3.2 SQLHandler.....	8
3.3.3 ResourceFactory.....	9
3.4 Updater.....	11
3.5 Verkkonäkymä.....	12
4.Tietokanta.....	14
5.Käyttöliittymä.....	15
5.1 DogmaMain.....	17
5.2 DogmaMenuComponent.....	17
5.3 DogmaSearchComponent.....	17
5.4 DogmaResourceTableComponent.....	18
5.5 DogmaURIComponent.....	18
5.6 DogmaFormView.....	19
6.Muut Komponentit.....	20
6.1 Tietosisältöluokat.....	20
6.1.1 DogmaResource.....	20
6.1.2 DogmaConnection.....	20
6.1.3 DogmaURI.....	20
6.2 JDBC-rajapinta.....	20
6.3 Reader.....	20
6.4 Konfiguraatitiedosto.....	21
6.5 Loki.....	21
6.6 DogmaDBCommComponent.....	21

6.7 DogmaClipboard.....	21
6.8 Ulkoiset komponentit.....	21
6.8.1 GUESS.....	21
6.8.2 HSQLDB.....	21
7. Testaussuunnitelma.....	23
7.1 Testauksen vaiheet.....	23
7.1.1 Yksikkötestaus.....	23
7.1.2 Integroititestaus.....	23
7.1.3 Järjestelmätestaus.....	23
7.2 Testattavat toiminnot.....	24
7.3 Testausympäristö.....	24
8. Liitteet.....	24

1. Johdanto

Dogma on Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojektin ryhmä. Tämä on ryhmän toteuttaman metadaperustaisen dokumenttiselaimen, Dogman, suunnitteludokumentti.

Dogmaa tullaan käyttämään lähinnä tutkimustarkoituksessa dokumenttien metadatan hallintaan ja dokumenttien välisten yhteyksien visualisointiin. Metadatan yhteyksien visualisointi ja tulkinta tapahtuu niiden tiedoista muodostettujen verkkojen avulla. Verkon avulla viitteitä voi selata eteenpäin niin paljon, kuin tietoa riittää tietokannassa ja linkkejä riittää resursseista toisiin.

Metadataa voi käsitellä myös ohjelman tarjoamassa taulukossa. Taulukosta voi valita uusia osajoukkoja, tutkia niitä verkossa ja tallentaa ne nimellä. Oleellista on, että käyttäjä pystyy tallentamaan tilanteensa ja jatkamaan seuraavalla kerralla samasta kohdasta.

Tässä dokumentissa esitetään järjestelmän komponentit rajapintoihin, kuvataan käyttöliittymää sekä esitellään testaussuunnitelma. Projektiin liittyvä materiaali on saatavilla ryhmän kotisivulta osoitteesta:

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

1.1 Sanasto

Attribuutti	Ominaisuusarvo.
GDF	GUESS:n käyttämä oma verkkojen tallennusformaatti.
GraphML	Verkkojen tallennusformaatti.
GUESS	The Graph Exploration System. Graafinen käyttöympäristö verkkojen visualisoimiseen ja selailemiseen.
HSQLDB	Javapohjainen SQL-tietokanta.
Jython	Java-integraatio Python-ohjelmointikielestä.
Metadata	Tietoa kuvaava tieto.
Resurssi	Metadasta tunnistettu yksilöllinen olio kuten henkilö, dokumentti tai julkaisukanava.
Resurssitaulukko	Ohjelmassa käytetty taulukko, johon on listattu käyttäjän hakuehdoilla löydetty tulosjoukko.
URI	Uniform Resource Identifier. Resurssin sijainnin yksilöllinen tunniste.
Verkkonäkymä	Resursseista muodostettu verkko GUESS:in esittämänä.

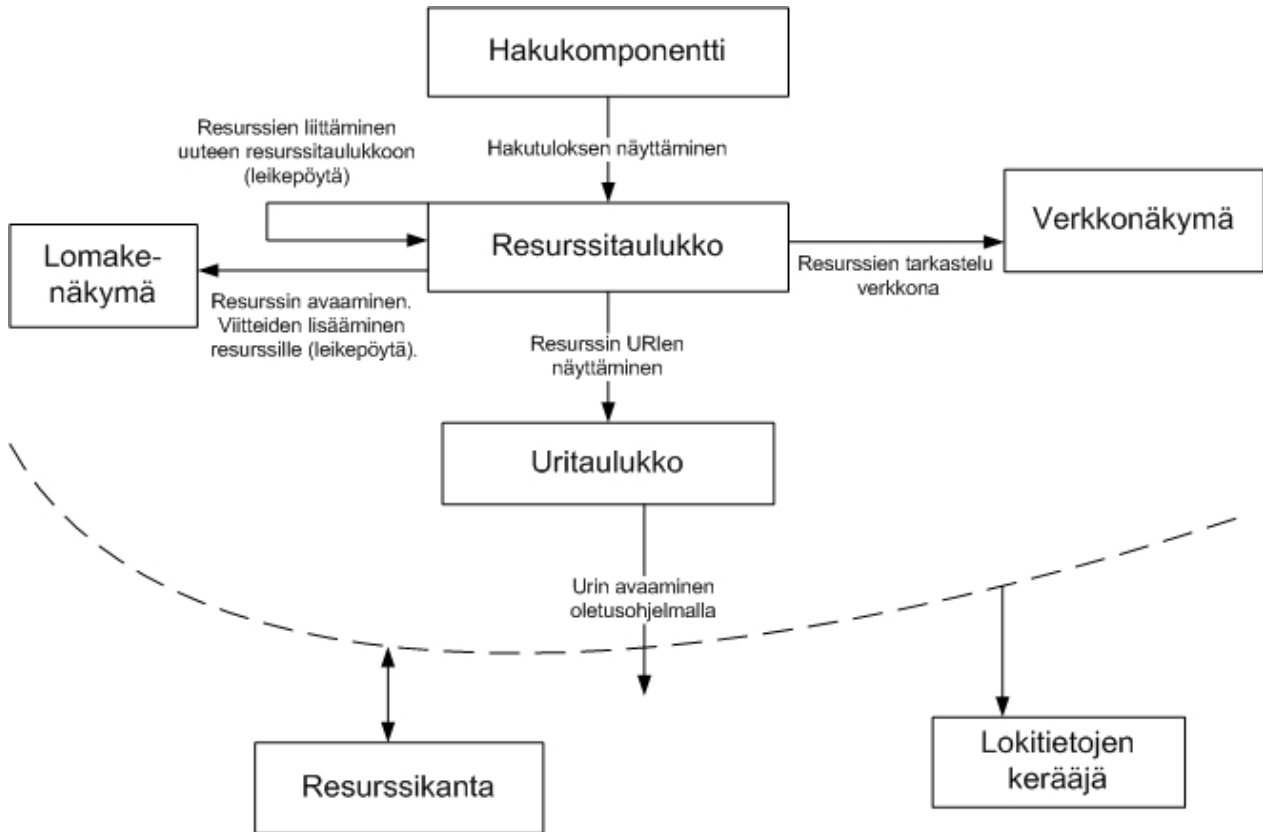
1.2 Dokumentin rakenne

Tässä luvussa eli luvussa 1 on esitelty lyhyt johdanto projektiin, suunnitteludokumentin sanasto sekä dokumentin rakenne. Luvussa 2 on esitelty ohjelmiston arkkitehtuuri kaavion ja pääkomponenttien muodossa. Luvussa 3 käydään läpi pääkomponenttien rajapinnat. Luvussa 4 tutkitaan Dogma – projektin tietokantaa ja sitä varten kehitettyä resurssikantakomponenttia. Käyttöliittymä on kuvattu luvussa 5, ja luvussa 6 käydään läpi muita komponentteja. Luvussa 7 on selitetty testaussuunnitelmaa.

2. Arkkitehtuurin kuvaus

Tässä luvussa kuvataan järjestelmää arkkitehtuurikaavion ja päätoimintojen avulla. Lisäksi luvussa on pääkomponenttien rajapintojen kuvaukset.

2.1 Arkkitehtuurikaavio



Kuva 1: Arkkitehtuurikaavio

Kuvassa esitetään ohjelman päätoimintoja.

2.2 Päätoimintojen kuvaukset

2.2.1 Resurssiselaimen avaaminen

Kun resurssiselain avataan, näkyy siinä viime kerralla auki olleet resurssit. Resurssiselaimesta voi luoda uuden instanssin valitsemalla uusi resurssiselain (New Browser). Resurssiselaimesta voi kopioida (ctrl-c, Copy) resursseja, ja ne voidaan liittää toiseen Resurssiselaimeseen valitsemalla liitä (ctrl-v, Paste).

2.2.2 Resurssien hakeminen hakutaulukon avulla

Käyttöliittymässä on hakutaulukko, johon voi syöttää arvoja attribuuteille. Jokainen taulukon sarakke kuvaa tietty attribuuttia ja jokainen rivi yhtä hakulauseen osaa. Jos taulukon ensimmäisellä rivillä attribuuttien arvoksi asetettaisiin $a_1 = 3$ ja $a_4 > 2$ ja toiselle riville $a_1 < 3$ ja $a_2 > 5$ olisi taulukosta saatava SQL-lause esimerkiksi muotoa

```
SELECT * FROM taulu WHERE (a1 = 3 AND a4 > 2) OR (a1 < 3 AND a2 > 5)
```

Tekstimuotoisille attribuuteille haku tehdään "like as" -tyyppisesti, ts. hakuarvon pienet ja suuret kirjaimet samastetaan, ja hakuarvon molemmille puolilla sallitaan olevan mitä tahansa, "*[hakuarvo]*".

Toinen tapa hakea resursseja on haku tagi(e)n perusteella. Tageille varattuun tekstikenttään kirjoitetut tagit määräävät hakua siten, että hakutuloksen resursseilla on oltava vähintään yksi näistä tageista. Hakea voi siis hakutaulukolla, tageilla, tai molemmilla.

Alussa hakutaulussa olisi vain yksi tyhjä rivi, mutta rivien määrä lisääntyy sitä mukaa kuin niitä täytetään. Valittaessa toiminto hae (Search), hakutaulukko lähettää tekemänsä SQL-lauseen resurssikannalle, joka suorittaa lauseen ja palauttaa tulosjoukon. Mikäli tagikenttään oli kirjoitettu tageja, otetaan leikkaus hakutaulukon ja tagihaun antamista tuloksista. Tulosjoukko näytetään resurssiselaimen resurssitaulukossa. Tulosjoukon kokoa voi rajoittaa käyttöliittymässä olevalla valintatoiminnolla.

2.2.3 Resurssien järjestäminen, hakeminen ja selaaminen

Resurssit talletetaan kaksiuolotteiseen String-matriisiin, jossa sarakkeet kuvaavat attribuutteja ja rivit resursseja. Käyttöliittymässä on hakutaulu, johon käyttäjä voi syöttää arvoja attribuuteille, joiden perusteella tietokannasta suoritetaan SQL-kysely. SQL-kyselyn tulokset näkyvät resurssiselaimessa ja resurssiselaimen attribuuttipalkkeja, joihin on kytketty tapahtumankuuntelijat, painamalla käyttäjä voi järjestää resurssit kunkin attribuutin mukaan nousevaan tai laskevaan järjestykseen.

Käyttäjä voi säätää sarakkeen leveyttä hiiren avulla. Resurssitaulun sarakkeiden lukitusta ja attribuuttirivien järjestystä ja lukitusta ei ehditä toteuttamaan. Resurssitaulukon sisäistä tekstihakua ei ehditä toteuttamaan.

Resurssiselaimessa on myös listaus viimeisille hauille, jolloin hakuja voidaan toistaa nopeasti.

2.2.4 Resurssien valitseminen resurssitaulukossa

Resurssitaulukossa voi valita resursseja käyttämällä erilaisia pikavalintoja. Resursseja voi valita useita ctrl-nappulaa pohjassa painamalla ja klikkaamalla hiirellä. Tietyllä hetkellä valitut resurssit erottuvat vaaleansinisinä riveinä. Shift pohjassa kursorin asettaminen tietylle riville lisää kerralla kaikki edellisen kursoririvin ja valitun kursoririvin väliset resurssit mukaan valittuihin. Kuitenkin jos kaikki tällä välillä kuuluvat jo valintoihin, ne poistetaankin valituista.

Valintoja voi tehdä myös näppäimistön kautta. Mahdollisia tapoja valita ovat valitse kaikki (select all, ctrl-a) ja poista valinnat (deselect, ESC).

Valitut resurssit voi asettaa leikepöydälle kopioimistoiminnolla (ctrl-c). Liittäminen leikepöydältä tapahtuu toiminnolla ctrl-v.

2.2.5 Lomakenäkymä ja resurssien muokkaus

Käyttäjä voi resurssitaulussa avata yksittäisen resurssin lomakenäkymään kaksoisklikkaamalla kyseistä resurssia. Lomakenäkymässä voidaan muokata resurssiin liittyviä viitteitä sekä resurssin attribuuttien arvoja. Resurssin avaus lomakenäkymään onnistuu myös Enter:iä painamalla rivin kohdistuksen ollessa resurssitaulukon rivillä. Jos lomakenäkymä on jo avoinna, klikkaamalla hiiren vasenta nappia resurssitaulun rivillä, kyseisen rivin resurssin tiedot näytetään lomakenäkymässä.

Monet resurssin attribuutit ovat muokattavissa. Jos attribuutin arvo ei ole muokattavissa, on se automaattisesti luotu tai automaattisesti laskettava. Jos attribuuttia on mahdollista

muokata, muokkaus onnistuu lomakenäkymän kautta. Attribuuttitiedot esitetään lomakenäkymässä taulukossa, jossa taulukon ensimmäisellä sarakkeella ovat attribuuttien nimet, ja toisella sarakkeella nimiä vastaavat attribuuttiarvot.

Viitteet esitetään lomakenäkymässä taulukkona, jossa ensimmäinen sarake kuvaa suuntaa, toinen viitteen toisen pään resurssin nimeä, ja kolmas viitteen tyyppiä. Lomakenäkymässä voi muuttaa resurssin viitteitä usella tavalla. Viitteen suuntaa voi muuttaa klikkaamalla suuntaa kuvaavaa nuolta. Viitteen tyyppiä voi muuttaa kirjoittamalla tekstikenttään uuden tyyppin. Viitteitä voi poistaa valitsemalla joukon rivejä ja painamalla sen jälkeen "Remove selected"-nappia, jolloin valitut viitteet poistuvat. Viitteiden lisääminen onnistuu kopiomalla joukon resurssitaulun rivejä (ts. resursseja) leikepöydälle, ja painamalla lomakenäkymän "Add"-nappia, jolloin lomakenäkymään lisätään viitteitä siten, että viitteen toisen pään resurssi on leikepöydällä oleva resurssi (suunta on oletusarvoisesti lomakenäkymän resurssista pois päin). Mikäli kyseinen resurssi oli jo viitteiden joukossa, viitettä ei lisätä.

Resurssin tagit ovat näkyvissä omassa tekstikentässään erotinmerkeillä eroteltuina, niiden ollessa suoraan muokattavissa.

Lomakenäkymään tehdyt muutokset tallennetaan vain, mikäli käyttäjä painaa "Save"-nappia. Tällöin resurssin attribuuttitiedot asetetaan lomakenäkymän tekstiarvojen perusteella (jos tekstiarvo ei ole yhteensopiva attribuuttityypin kanssa, muutosta ei tallenneta). Viitteiden tallentumisessa on huomioitavaa, että muutokset päivittyvät myös viitteen toiseen resurssiin. "Reset"-nappia painamalla lomakenäkymään tehdyt muutokset (viimeisimmän resurssin tallentamisen jälkeen) poistetaan.

2.2.6 Uuden resurssin luominen

Käyttäjä voi luoda uuden resurssin valitsemalla käyttöliittymästä toiminnon uusi resurssi ("New resource", ctrl-n). Toiminnon aktivoituessa käyttöliittymä luo uuden resurssin ja avaa sen lomakenäkymään, jossa se on muokattavissa.

Samalla käyttäjä voi myös luoda viitteitä toisiin resursseihin ja toisista resursseista sekä muokata attribuuttien arvoja, kuten kohdassa 2.2.5.

2.2.7 Resurssien yhdistäminen, puolittaminen ja poistaminen

Näitä toimintoja ei ehditty toteuttaa, toivottujen toimintojen kuvaukset ovat kuitenkin tässä jatkokehitystä varten.

Resurssiselaimessa voi yhdistää kaksi solmua yhdeksi. Yhdistys (Union) tapahtuu valitsemalla parent-solmu, johon toinen solmu ja kaikki sen kaaret yhdistetään. Resurssien puolitus (Split) tapahtuu valitsemalla resurssin viitteet ja siirtämällä ne uuteen resurssiin. Valitut viitteet eivät enää osoita vanhaan resurssiin.

Resurssit voidaan poistaa resurssiselaimen avulla valitsemalla poista (Delete) toiminto.

2.2.8 URI-taulukko ja toiminnot

Resurssiselaimessa on URI-taulukko, jossa näkyy valittuun resurssiin liittyvät URI:t ja niiden attribuutit. URI:en attribuutteja voi muokata taulukossa. Jos URI:a klikkaa hiirellä tai URI:n ollessa valittuna painaa Enter:iä, avautuu se tyyppinsä määräämässä oletusohjelmassa.

Uritaulukon voi myös järjestää attribuuttien mukaan klikkaamalla hiirellä attribuuttiotsikkoa.

2.2.9 Vapaamuotoinen SQL-kysely

Käyttöliittymästä voi valita syöteikkunan, johon voi kirjoittaa vapaamuotoisen SQL-kyselyn,

joka ajetaan tietokannassa. Vapaamuotoinen kysely on tarkoitettu lähinnä ylläpitoa varten. Kyselyn mahdollinen tulos tulostetaan uuteen ikkunaan.

2.2.10 Resurssien tarkastelu verkkonäkymässä

Resurssiselaimesta valitut resurssit voidaan avata verkkonäkymään tarkastelua varten ("Show resources in graph", ctrl-g). Verkkonäkymässä resurssit esitetään solmuina, ja niiden väliset yhteydet nuolina. Näkyvissä ovat (aluksi) resurssiselaimesta valitut resurssit, niitä klikkaamalla voidaan tuoda näkyville myös muita resursseja, joihin näistä resursseista on viitteitä. Tätä verkon laajentamista voidaan jatkaa tiettyyn, ennaltamääritettyyn syvyyteen asti. Myös uusia, resurssiselaimesta valittuja resursseja, voidaan lisätä jo olemassaolevaan verkkonäkymään. Tällöin verkossa näkyvät sekä vanhat, että uudet resurssit.

2.2.11 Viimeisimpien hakujen listaaminen

Resurssiselaimessa, hakutaulukon yläpuolella, on toimintoja haun suorittamista, tallentamista, ja vanhojen hakujen hakemista varten. Hakujen suorittamista on jo tarkasteltu edellisissä kohdissa. Kun haku suoritetaan painamalla "Search"-nappia, haku tallentuu nimettömänä, suoritusajankohdan perusteella. Hakuja voidaan tallentaa myös "Save search as"-napilla, jolloin napin vieressä olevan tekstikentän sisältö tallentuu haun nimeksi. Tällä hetkellä vanhoja hakuja voidaan hakea pudotusvalikosta nimen tai päivämäärän mukaan järjestettynä. Toistaiseksi hakuja ei voi tallentaa pysyvästi.

2.2.12 Tapahtumien tallentaminen loki-tiedostoon

Käyttöliittymässä tehtyjä tapahtumia tallennetaan loki-tiedostoon. Näitä ovat mm. resurssin esiintyminen hakutuloksessa, resurssin avaaminen lomakenäkymässä, tai resurssin attribuutin muuttaminen. Tapahtumat tallennetaan loki-tiedostoon tekstimuodossa.

2.2.13 Muut toiminnallisuudet lyhyesti

Resurssiselaimessa voi lisätä valitulle resurssijoukolle tagin menu toiminnolla "Add Tag", jolloin avautuu erillinen dialogi tagitekstin antamista varten.

2.3 Pääkomponentit lyhyesti

2.3.1 Käyttöliittymä

Käyttöliittymä kattaa graafisen käyttöliittymän komponentit; resurssitaulukon, uritaulukon ja hakukomponentin, lomakenäkymän sekä tapahtumankuuntelijat. Lähes kaikki Dogma-ohjelmiston toiminnot tapahtuvat käyttöliittymän kautta.

2.3.2 Loki

Loki-komponentti pitää kirjaa tehdyistä tapahtumista. Lokin voi tallentaa tekstitiedostoon ja sen voi myös avata tekstitiedostosta. Loki tarjoaa mahdollisuuden resurssin, aikavälin ja tapahtumien lukumäärään liittyviin hakuihin.

Tapahtumalokiin talletettavat tapahtumat ovat

- Resurssin attribuuttien muuttaminen
- Resurssin avaaminen lomakenäkymään
- Resurssin avaaminen oletusohjelmalla

- Resurssin esiintyminen hakutuloksessa
- Resurssin liittäminen osajoukkoon
- Resurssin liittäminen verkkonäkymään

2.3.3 Resurssikanta

Dogma tallentaa resurssit ja niihin liittyvät viitteet sekä URI:t resurssikantaan, josta muut palvelut käyttävät niitä. Resurssikanta on Dogman keskeisin komponentti.

2.3.4 Päivittäjä

Dogma lataa dokumentit käyttöönsä erilaisten lukijoiden avulla, jotka käyttäjä voi itse toteuttaa. Lukijat ovat yhteydessä päivittäjä - olioon, joka antaa tietoa resurssikannalle.

Päivittäjä tarkistaa resurssikannasta onko lisättävä resurssi, viite tai URI jo siellä ja pyrkii välttämään tiedon moninkertaista tallennusta. Jos päivittäjä pitää kirjaa jo luetuista lähteistä, esimerkiksi tiedostoista tai hakemistoista, ei sitä lukija-komponenttien tarvitse tätä erikseen tehdä.

2.3.5 Verkkonäkymä

Verkkonäkymä toteutetaan GUESS-komponentin avulla. Käyttäjällä on resurssiselaimen kautta (Kun resursseja valittu -> Avaa verkkonäkymässä) mahdollisuus avata verkkonäkymässä näkyviä solmuja (resursseja), jolloin hän pystyy tutkimaan niiden välisiä yhteyksiä.

Verkkonäkymä avataan käyttöliittymän kautta ja se saa resurssiselaimelta alkuarvona epätyhjän joukon solmuja, jotka se piirtää käyttäjälle näkyviin.

3. Pääkomponenttien rajapinnat

Alla on listattu pääkomponenttien rajapinnat ja kerrottu niiden toiminnasta niiden tarjoamien palveluiden avulla.

3.1 Käyttöliittymä

Käyttöliittymä käyttää kaikkia muita komponentteja, eikä sille voi ryhmän taitojen puitteissa (vielä) varsinaisesti kehittää rajapintaa.

3.2 Loki

Loki-komponenttiin kirjataan tehdyt tapahtumat aikamerkinnän kanssa myöhempää analysointia varten. Katso luku 2.3.2 tapahtumien listausta varten.

Palvelun nimi	addEvent
Palvelun kuvaus	Lisää tapahtuman lokioliioon.
Sisäänotettavat parametrit	iResource r, mihin resurssiin lisäys viittaa. String event, tapahtuman kuvaus.
Ulos annettavat tulokset	-

Poikkeustilanteet	Jos joku annetuista parametreista null tai poikkeava, tallennetaan tapahtuma silti. Jos tiedostoon ei voida tallentaa, ilmoitetaan siitä.
Alkuehdot	Loki-olio alustettu.
Loppuehdot	Tapahtuma annetuilla parametreilla on lokioliassa

3.3 ResourceBase

Muu Dogma-komponentti lähettää tälle luokalle kyselynsä ja riippuen kyselyn laadusta käynnistää ResourceBase-kyselyistä saadun tuloksen iteroinnin tai muun tarvittavan operaation. ResourceBasen kautta hoidetaan uuden tiedon luominen saatujen parametrien mukaan sekä palautetaan käyttöliittymälle tietoja iResource-, iUri- sekä iConnection-rajapinnan toteuttavien olioiden muodossa.

ResourceBase on jaettu useampaan alikomponenttiin toimintojensa perusteella. Jokainen alikomponentti on vaihdettavissa ilman että ohjelman toiminta häiriintyy.

3.3.1 BaseConnection

BaseConnection on alin tietokannanhallinnan taso, joka hoitaa yhteyden muodostuksen ja kyselyt. BaseConnection – olion toteuttamat palvelut on listattu tauluoliona.

Palvelun nimi	BaseConnection
Palvelun kuvaus	Luo BaseConnection-olion, joka hoitaa yhteyden muodostuksen ja kyselyt.
Sisäänotettavat parametrit	String jdbc, käytettävän jdbc-ajurin nimi
Ulos annettavat tulokset	BaseConnection-olio
Poikkeustilanteet	Jos jdbc-ajurin tietoa ei löydy, käytetään oletusajuria (HSQLDB). Jos oletusajuriakaan ei löydy palautetaan null-olio.
Alkuehdot	Konfiguraatitiedostossa on käytettävän jdbc-ajurin nimi
Loppuehdot	BaseConnection-olio luotu, yhteys tietokantaan muodostettu.

Palvelun nimi	sqlQuery
Palvelun kuvaus	Suorittaa sql-kyselyn kantaan jo alustetulla yhteydellä.
Sisäänotettavat parametrit	String sql, ajettava sql-kysely
Ulos annettavat tulokset	ResultSet-olio, jossa kyselyn tulokset
Poikkeustilanteet	Jos yhteyttä kantaan ei löydy tai jokin

	muu sql-virhe tapahtuu palauttaa null.
Alkuehdot	Yhteys tietokantaan muodostettu, eli <code>BaseConnection</code> – olio on luotu.
Loppuehdot	Ajetun kyselyn tulokset <code>ResultSet</code> -oliossa, jota voidaan käydä läpi iteroimalla.

Palvelun nimi	<code>closeConnection()</code>
Palvelun kuvaus	Sulkee tietokantayhteyden.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	-
Poikkeustilanteet	Jos yhteyttä kantaan ei löydy ei tehdä mitään.
Alkuehdot	Yhteys tietokantaan muodostettu.
Loppuehdot	Tietokantayhteys suljettu.

3.3.2 SQLHandler

SQLHandler jäsentää annetun luokan sen muuttujien arvojen perusteella kyselyksi. Annettujen luokkien on toteutettava rajapinta `iResource`. Tämä luokka sisältää vain staattisia metodeja.

Palvelun nimi	<code>resourceToQuery</code>
Palvelun kuvaus	Muuttaa parametrina annetun resurssin tietokantahauksi SQL-formaatissa.
Sisäännotettavat parametrit	<code>iResource resource</code> , resurssiolio joka halutaan tietokantahauksi
Ulos annettavat tulokset	<code>String</code> – olio, jossa sql-kysely
Poikkeustilanteet	Jos syöteolio null, palautetaan null.
Alkuehdot	Syötettävän <code>resource</code> -olion pitää toteuttaa <code>iResource</code> – rajapinta.
Loppuehdot	Resurssin tiedoista luotu sql-muotoinen kysely.

Palvelun nimi	<code>UriToQuery</code>
Palvelun kuvaus	Muuttaa parametrina annetun URIn tietokantahauksi SQL-formaatissa.
Sisäännotettavat parametrit	<code>iUri uri</code> , uri-olio joka halutaan tietokantahauksi

Ulos annettavat tulokset	String – olio, jossa sql-kysely
Poikkeustilanteet	Jos syöteolio null, palautetaan null.
Alkuehdot	Syötettävän uri-olion pitää toteuttaa iUri – rajapinta.
Loppuehdot	Urin tiedoista luotu sql-muotoinen kysely.

Palvelun nimi	ConnectionToQuery
Palvelun kuvaus	Muuttaa parametrina annetun yhteyden tietokantahauksi SQL-formaatissa.
Sisäännotettavat parametrit	iConnection conn, yhteys-olio joka halutaan tietokantahauksi
Ulos annettavat tulokset	String – olio, jossa sql-kysely
Poikkeustilanteet	Jos syöteolio null, palautetaan null.
Alkuehdot	Syötettävän yhteys-olion pitää toteuttaa iConnection – rajapinta.
Loppuehdot	Yhteyden tiedoista luotu sql-muotoinen kysely.

Palvelun nimi	dataToQuery
Palvelun kuvaus	Muuttaa parametrina annetun dataolion tietokantahauksi SQL-formaatissa. Erilaisia käsiteltäviä olioita voi olla useita.
Sisäännotettavat parametrit	iData data, dataolio joka halutaan tietokantahauksi
Ulos annettavat tulokset	String – olio, jossa sql-kysely
Poikkeustilanteet	Jos annetun dataolion muutos tietokantahauksi epäonnistuu, palautetaan null. Jos syöteolio null, palautetaan null.
Alkuehdot	Syötettävän data-olion pitää toteuttaa iData – rajapinta.
Loppuehdot	Annetusta dataoliosta muodostettu sql-muotoinen kysely.

3.3.3 ResourceFactory

ResourceFactory luo saapuneiden pyyntöjen perusteella uusia iResource-rajapinnan toteuttavia olioita, joita voidaan käyttää hakemaan tai näyttämään resurssin tietoja. Sen konstruktorille annetaan parametrina hakutulos (ResultSet – olio), jonka ResourceFactory jäsentää riveiksi ja hakee tietoja niiden perusteella.

Palvelun nimi	ResourceFactory
Palvelun kuvaus	Luo uusia iResource – rajapinnan toteuttavia resurssiolioita annettujen hakutulosten perusteella.
Sisäännotettavat parametrit	ResultSet result, sql-kyselyn tulos
Ulos annettavat tulokset	ResourceFactory-olio
Poikkeustilanteet	result – olio virheellinen tai tyhjä.
Alkuehdot	Syöte korrekti ResultSet-olio
Loppuehdot	ResourceFactory – olio alustettu.

Palvelun nimi	getNextResource
Palvelun kuvaus	Hakee tulosjoukosta seuraavan resurssin ja jäsentää siitä iResource-olion.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	iResource – olio
Poikkeustilanteet	Jos tulosjoukossa ei seuraavaa iResource-oliota palautetaan null.
Alkuehdot	ResourceFactory-olio luotu.
Loppuehdot	Jos resursseja jäljellä, niin resurssialkioiden määrä ResourceFactory - oliossa vähenee yhdellä.

Palvelun nimi	getNextConnection
Palvelun kuvaus	Hakee tulosjoukosta seuraavan yhteyden ja jäsentää siitä iConnection-olion.
Sisäännotettavat parametrit	-
Ulos annettavat tulokset	iConnection – olio
Poikkeustilanteet	Jos tulosjoukossa ei seuraavaa iConnection-oliota palautetaan null.
Alkuehdot	ResourceFactory-olio luotu.
Loppuehdot	Jos yhteyksiä jäljellä, niin yhteysalkioiden määrä ResourceFactory - oliossa vähenee yhdellä.

Palvelun nimi	getNextUri
Palvelun kuvaus	Hakee tulosjoukosta seuraavan URIn

	ja jäsentää siitä iUri-olion.
Sisäänottavat parametrit	-
Ulos annettavat tulokset	iUri – olio
Poikkeustilanteet	Jos tulosjoukossa ei seuraavaa iUri-oliota palautetaan null.
Alkuehdot	ResourceFactory-olio luotu.
Loppuehdot	Jos ureja jäljellä, niin urialkioiden määrä ResourceFactory - oliossa vähenee yhdellä.

3.4 Updater

Päivittäjä ottaa vastaan tietoja eri lukijoilta ja jäsentää tiedot tietokannalle sopivaan muotoon. Tietojen prosessoinnin jälkeen päivittäjä lisää tiedot ResourceBasen välityksellä tietokantaan.

Palvelun nimi	update
Palvelun kuvaus	Lähetää tehdyt muutokset ResourceBase-oliolle.
Sisäänottavat parametrit	-
Ulos annettavat tulokset	Updater-olio
Poikkeustilanteet	ResourceBase-oliota ei ole olemassa, operaatio keskeytetään.
Alkuehdot	Muutoksia tehty.
Loppuehdot	Muutokset lähetetty ResourceBase-oliolle.

Palvelun nimi	processResource
Palvelun kuvaus	Tarkastaa onko Resurssitietokannassa olemassa jo syötteenä saatu resurssi, kanonisoidun nimen toimiessa tunnisteena, ja lisää sen sinne tarvittaessa.
Sisäänottavat parametrit	iResource resource
Ulos annettavat tulokset	-
Poikkeustilanteet	-
Alkuehdot	Updater – olio luotu, lukija-olio on lukenut dataa.
Loppuehdot	Resurssitietokannasta löytyy lisätty resurssiolio.

Palvelun nimi	processConnection
Palvelun kuvaus	Tarkastaa onko Resurssitietokannassa

	olemassa jo syötteenä saatu yhteys ja lisää sen sinne tarvittaessa.
Sisäännotettavat parametrit	iConnection connection
Ulos annettavat tulokset	-
Poikkeustilanteet	-
Alkuehdot	Updater – olio luotu, lukija-olio on lukenut dataa.
Loppuehdot	Resurssitietokannasta löytyy lisätty yhteysolio.

Palvelun nimi	processUri
Palvelun kuvaus	Tarkastaa onko Resurssitietokannassa olemassa jo syötteenä saatu URI ja lisää sen sinne tarvittaessa.
Sisäännotettavat parametrit	iUri uri
Ulos annettavat tulokset	-
Poikkeustilanteet	-
Alkuehdot	Updater – olio luotu, lukija-olio on lukenut dataa.
Loppuehdot	Resurssitietokannasta löytyy lisätty URI.

3.5 Verkkonäkymä

Verkkonäkymä toteutetaan valmiin ohjelman, GUESS:in, avulla. Dogman ja Guessin välillä toimii rajapintana DogmaGraphHandler. DogmaGraphHandlerin tehtäviin kuuluu gdf-tiedostojen kirjoittaminen, GUESS:in käynnistäminen ja resurssien lisääminen GUESS:in verkkonäkymään.

Palvelun nimi	addResource
Palvelun kuvaus	Käynnistää GUESS-komponentin ja lisää siihen annetut resurssit. Jos GUESS-komponentti on jo käynnissä, ainoastaan lisätään resurssit r.
Sisäännotettavat parametrit	Vector<DogmaResource> r, resurssivektori
Ulos annettavat tulokset	-
Poikkeustilanteet	Resurssi jo valmiiksi verkossa. Resurssin naapureita haettaessa ei saada yhteyttä tietokantaan / ko. resurssia ei löydy tietokannasta.

Alkuehdot	DogmaGraphHandler-instanssi haettu
Loppuehdot	Kirjoitetaan alkuperäinen resurssijoukko gdf-tiedostoon ja käynnistetään GUESS sen avulla, tai vaihtoehtoisesti, jos GUESS on jo käynnistetty, lisätään ainoastaan resurssit verkkonäkymään.

Palvelun nimi	setResourceBase
Palvelun kuvaus	Asettaa DogmaGraphHandlerin käyttämään parametrina annettua ResourceBasea.
Sisäänotettavat parametrit	ResourceBase rb, käytettävän tietokantaluokan ilmentymä
Ulos annettavat tulokset	-
Poikkeustilanteet	Annettu parametri on null, tällöin tietokantaa ei aseteta
Alkuehdot	DogmaGraphHandler-instanssi haettu
Loppuehdot	Uusi resurssikanta asetettu

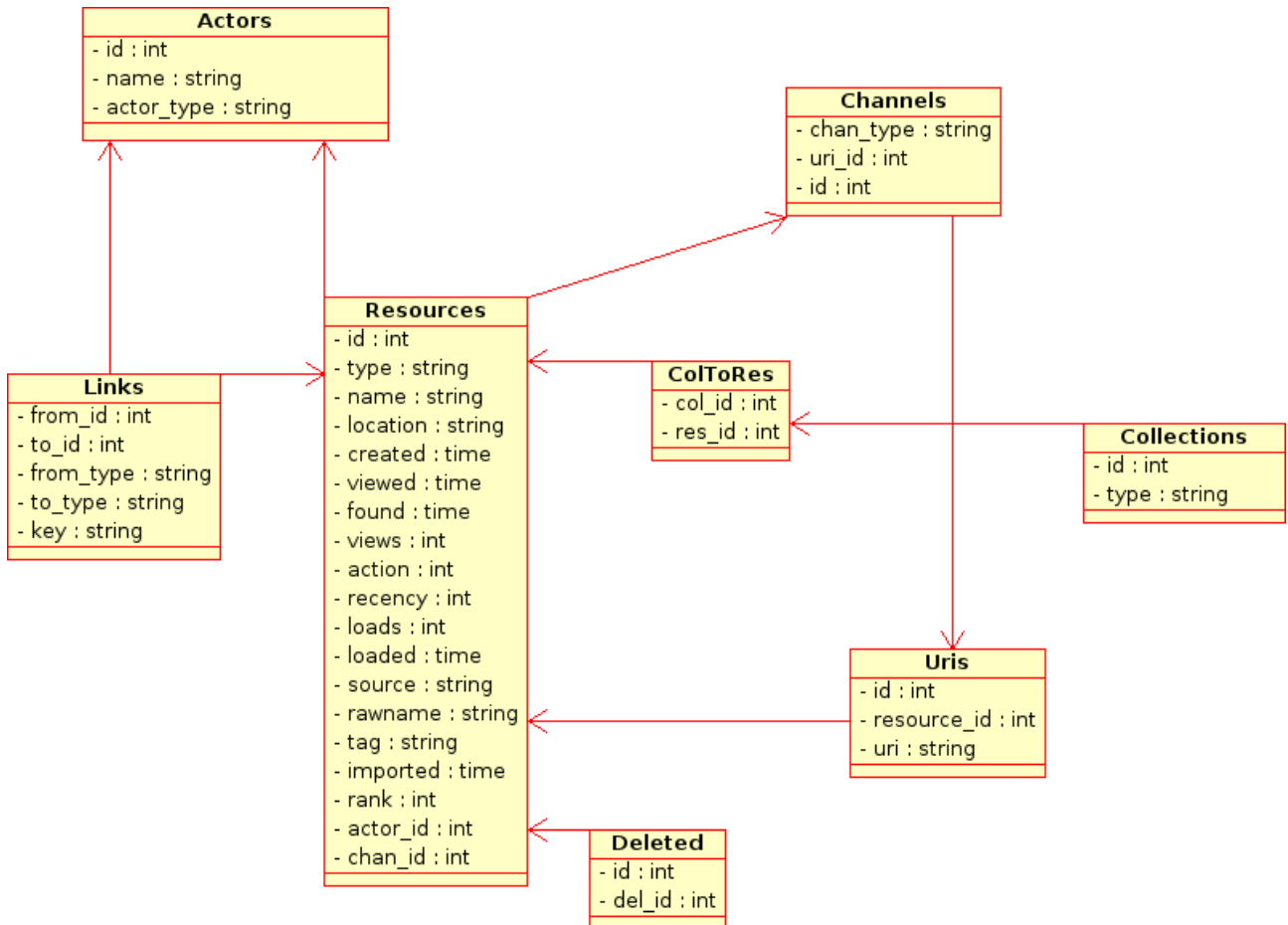
Palvelun nimi	setGraphWalkDistance
Palvelun kuvaus	Asettaa etäisyyden, jota käytetään kirjoittaessa alkuperäisten parametreina annettujen resurssien naapureita gdf-tiedostoon. Naapuresursseja haetaan n:n askeleen päähän.
Sisäänotettavat parametrit	int distance, uusi etäisyys
Ulos annettavat tulokset	-
Poikkeustilanteet	distance < 0, tällöin säilytetään vanha arvo
Alkuehdot	DogmaGraphHandler-instanssi haettu
Loppuehdot	Uusi etäisyys asetettu

4. Tietokanta

Tietokantayhteys hoidetaan JDBC-rajapinnan kautta määriteltyyn HSQLDB-tietokantaan. Tietokanta on oletusarvoisesti tiedosto, joka sijaitsee Dogman juurihakemistossa ja on nimeltään dogma.db. Tietokantayhteys voidaan myös alustaa käyttämään erinimistä tai muunlaista HSQLDB-tietokantaa.

Tietokannan tulee kuitenkin olla valmiiksi alustettu, että ohjelma toimii oikein. Alustuksella tarkoitetaan, että sieltä on löydyttävä tarvittavat taulut.

Tietokannan rakenne on seuraava:



Kuva 2: Tietokannan rakenne

Taulujen kuvaukset:

Taulun nimi	Resources
Kuvaava luokka ohjelmistossa	DogmaResource
Lyhyt kuvaus	Resources kerää dokumenttien statistista tietoa, dokumentin nimen sekä identifioijan resurssiin liittyvään toimijaan sekä URIin. Yksittäisiä resursseja haetaan yksilöllisen canonamen perusteella.

Taulun nimi	Actor
--------------------	-------

Kuvaava luokka ohjelmistossa	DogmaResource
Lyhyt kuvaus	Actor-aulussa ovat ne toimijat, jotka tuottavat resursseja. Toimija voi olla henkilö, tutkimusryhmä tai -laitos tai yliopisto. Toimijat voivat olla keskenään hierarkisessa suhteessa, esimerkiksi tutkimusryhmä on osa jotakin yliopistoa jne.

Taulun nimi	Channel
Kuvaava luokka ohjelmistossa	DogmaResource
Lyhyt kuvaus	Channel-tiluun kerätään kanavia, joiden kautta löydetään uusia osoitteita. Tämä taulu on pitkälti tarkoitettu auttamaan

Taulun nimi	References
Kuvaava luokka ohjelmistossa	DogmaConnection
Lyhyt kuvaus	References-tiluussa pidetään yllä tietoa eri resurssien välisistä suhteista, esimerkiksi kuka on tuottanut tietyn resurssin, onko jokin toimija osa jotain toista toimijaa (kts yllä) jne. Referenssejä käytetään muun muassa muodostamaan verkkonäkymää eri resursseista.

Taulun nimi	Uris
Kuvaava luokka ohjelmistossa	DogmaURI
Lyhyt kuvaus	Uris-tiluussa säilytetään tiedot kaikkien resurssien URI-tiedoista. Mikäli jotakin URIa halutaan hakea, on sitä enne haettava URIa käyttävän resurssin tunniste (id).

Taulun nimi	Deleted
Kuvaava luokka ohjelmistossa	-
Lyhyt kuvaus	Deleted-tiluussa ylläpidetään tietoa niistä resursseista, joita ei näytetä resurssitaulussa. Mikäli varsinainen poisto suoritetaan, tästä taulusta katsotaan, mitkä resurssit poistetaan pysyvästi kannasta. Oletusarvoisesti tätä ei tehdä.

5. Käyttöliittymä

Pääkäyttöliittymä, DogmaMain, koostuu neljästä käyttöliittymäkomponentista.

- (1) Menukomponentti, DogmaMenuComponent, joka sisältää menupalkin ja sen kuuntelijat
- (2) Hakukomponentti, DogmaSearchComponent, jossa on hakutaulu ja haku tageille.
- (3) Resurssitaulukomponentti, DogmaResourceTableComponent, jossa resurssit näytetään
- (4) URItaulukomponentti, DogmaURIComponent, jossa näytetään aina valitun resurssin urit.

The screenshot shows the Dogma application window with the title "Dogma. That's all you need to know!". The interface includes a menu bar (File, Database, Help), a search bar with a "Search" button, and a "Sort by" dropdown menu with options for "name" and "time". There are also buttons for "New search" and "Save search as".

Below the search bar, there are two tables. The first table is empty and has a red "2" in the center. The second table is a resource table with the following data:

resourceClass	type	name	connectionsFrom	connectionsTo	connections
actor	Organization	ACADEMY OF MATHE...	1	1	2
actor	Person	ABERER K	0	1	1
actor	Person	ABITEBOUL S	0	1	1
actor	Person	ACHARYYA S	0	1	1
actor	Person	ALON N	0	1	1
actor	Person	ALTMAN A	1	2	3
actor	Person	ALVIS A	1	1	2
actor	Person	AMGOUD L	0	1	1
actor	Person	ANDERSON R	0	1	1
actor	Person	AUGUST	0	1	1
actor	Person	AUTHORS C	1	1	2
resource	Article	A BRIEF INTRODUCTI...	1	0	1

A red "3" is placed in the "connectionsFrom" column of the row for "ALTMAN A". Below the resource table is a table for "MyURI" with columns "MyURI", "Type", "Name", and "Location". This table is empty and has a red "4" in the center.

Näiden komponenttien lisäksi käyttöliittymään kuuluu lomakenäkymä, FormView, johon resurssit voidaan avata tarkempaa tarkastelua varten. Käyttöliittymässä tehdyt muutokset ja lisäykset tehdään DogmaDBCommComponent – luokan välityksellä tietokantaan. DogmaDBCommComponent – komponentista voi tietenkin kutsua myös tietokantaa suoraan, jolloin tietokantaa enemmän tarvitsevat komponentit pääsevät siihen käsiksi.

Jos käyttöliittymää haluaa ajatella Model-View-Controller – mallin avulla, voidaan jokainen komponentti ajatella omana Model-View-Controller – palasena. DogmaMain – luokassa vain lisätään kaikki palat yhteen.

Komponenttien työstämisen ja DogmaMain – luokan tekemän työn minimoimiseksi tekee jokainen komponentti kuitenkin oman asettelunsa DogmaMain – luokan tarjoamaan Shell – olioon.

Tässä kappaleessa tarkastellaan enemmän näiden komponenttien toimintaa ja yhteistyötä.

5.1 DogmaMain

DogmaMain on pääluokka jossa ohjelma avataan ja siihen liittyvät komponentit luodaan. Kun pääluokka alustetaan, luodaan neljä siihen kuuluvaa käyttöliittymäkomponenttia ja tietokantayhteys.

Käyttöliittymäkomponentit saavat myös luomisparametreinaan tiedon tietokantakomponentista, jonka avulla tietoon pääsee käsiksi.

DogmaMain – luokka tarjoaa kaksi tärkeää funktiota muille komponenteille

```
public void setUriComponentContent(DogmaResource r)
```

Lähetää urikomponentille resurssin, johon liittyvät urit se näyttää.

```
public void setResourceTableContent(Vector<DogmaResource> v)
```

Lähetää resurssitaulukomponentille resurssivektorin, jossa olevat resurssit näytetään resurssitaulussa.

DogmaMain pitää myös Lomakenäkymä (FormView) – luokan kanssa kirjaa auki olevista instansseista, jolla mahdollistetaan tietokannan oikein sulkeutuminen.

5.2 DogmaMenuComponent

DogmaMenuComponent on GUI:n komponentti, joka luo menupalkin ja sen sisällön ohjelmaa varten. Alla lyhyesti listattuna menun toiminnot.

File

New Browser, avaa uuden dogmamain - ikkunan.

New Resource, avaa lomakenäkymän uuden resurssin lisäämistä varten.

Exit, sulkee kyseisen DogmaMain - ikkunan.

Database

Import Textfile, kysyy käyttäjältä jonkun updaterin luomaa dump-tiedostoa, jonka sisältö lisätään auki olevaan kantaan.

Run SQL-Query, ajaa vapaamuotoisen SQL-kyselyn tietokannassa ja pyrkii kertomaan jotain tuloksesta.

Save Modifications, tallentaa DogmaDBCommComponenttiin tehdyt muutokset heti kantaan. Tämä tapahtuu myös ohjelmaa suljettaessa.

Help

About, näyttää hieman tietoa ohjelmasta

User Guide, avaa käyttöohjeen DogmaOpen – luokan avulla käyttöjärjestelmässä olevaan internetselaimeen.

5.3 DogmaSearchComponent

DogmaSearchComponent on GUI:n komponentti, joka luo hakualueen ja sen sisällön. Hakualue koostuu hausta tageille ja resurssin attribuuteille. Lisäksi alueella on valikko vanhan haun toistamista varten.

DogmaSearchComponenttia luodessa luodaan sille hakutaulu, DogmaSearchTable, johon käyttäjä voi lisätä hakuehtoja ja joka lisää hakuehdoille tyhjiä rivejä aina tarvittaessa. Alueelle lisätään myös erillinen tekstikenttä tagien hakua varten.

Search – nappia painettaessa hakukomponentti hakee tietokannasta kaikki hakuehtoihin liittyvät resurssit. Jos hakuehdot ovat tyhjiä, palauttaa tietokanta kaikki resurssit. Jos sekä tagikenttään, että hakutauluun on asetettu ehtoja, haetaan kannasta kaikki resurssit jotka sopivat kumpiinkin ehtoihin. Kun resurssien haku on toteutettu, hakukomponentti antaa tietokannalta saadun resurssivektorin DogmaMain – luokan ”setResourceTableContent” - metodin kautta resurssitaulukomponentille.

5.4 DogmaResourceTableComponent

DogmaResourceTableComponent on GUI:n komponentti, joka luo resurssitaulukon ja näyttää sen sisällön. Komponentissa käytetään DogmaTable – taulukkoa, joka on muokattu versio vapaasti jaossa olevasta KTable – komponentista.

Komponentti saa näytettävät resurssit DogmaMain – luokassa olevan metodin kautta resurssivektorina, jonka sisällä olevat resurssit piirretään taulukkoon.

Resurssitaulun päällä voi avata oikeaa hiiren näppäintä painamalla pikavalikon, jonka toiminnoille on myös pikanäppäimet:

- Ctrl + a, valitsee kaikki taulukossa olevat resurssit.

- Ctrl + b, avaa valitut resurssit uudessa DogmaMain – ikkunassa.

- Ctrl + c, kopioi valitut resurssit leikepöydälle.

- Ctrl + e, avaa valitun resurssin editoimista varten lomakenäkymään (DogmaFormView).

- Ctrl + g, näyttää valitut resurssit verkkonäkymässä.

- Ctrl + n, avaa uuden resurssin editoimista varten lomakenäkymään (DogmaFormView).

- Ctrl + t, avaa ikkunan jossa voi lisätä tagin valituille resursseille.

- Ctrl + v, liittää leikepöydällä olevat resurssit aktiiviseen DogmaMain – ikkunaan.

Jos taulukosta valitaan resurssi, lähetetään se DogmaMain – luokan ”setURITableContent” - metodin kautta uritaulukomponentille.

5.5 DogmaURIComponent

DogmaURIComponent on GUI:n komponentti, joka luo uritaulukon ja näyttää sen sisällön. Urien näytössä käytetään DogmaTable – taulukkoa, joka on muokattu versio vapaasti jaossa olevasta KTable – komponentista.

Tauluun voi lisätä myös uusia ureja, jotka tallennetaan aina aktiivisena olevaan resurssiin. Urit voidaan myös avata oletusohjelmalla joko tuplaklikkaamalla urin riviä, tai valitsemalla oikeaa hiirennappia painamalla avautuvasta menusta ”Open URI in default program.”. Avaamisessa käytetään DogmaOpen – luokkaa, joka tarkastaa käytettävän käyttöjärjestelmän ja pyrkii löytämään oikean ohjelman urille.

Komponentissa tehdyt muutokset tallennetaan DogmaDBCommComponenttiin, joka

ohjelmaa suljettaessa tallentaa muutokset kantaan.

5.6 DogmaFormView

DogmaFormView on erillinen komponentti, jossa voidaan näyttää resurssiin liittyvät tiedot ja sen yhteydet muihin resursseihin.

Komponentti on toteutettu TabFolder – luokan avulla, jolloin yhdellä sivulla on resurssin attribuutit ja siihen liittyvät tagit. Toisella sivulla on taas resurssiin liittyvät yhteydet ja niiden tiedot.

Komponentissa tehdyt muutokset voidaan tallentaa kantaan valitsemalla ”Save”. Yhteyksien lisääminen toimii leikepöydän kautta. Jos käyttäjä on kopioinut DogmaMain – ikkunassa resursseja leikepöydälle, voidaan ne liittää lomakenäkymään ”Add” - painikkeella. Muutokset tallennetaan sekä DogmaDBCommComponent – luokkaan, että sen avulla suoraan tietokantaan nopeiden jatkooperaatioiden mahdollistamiseksi.

DogmaFormViewin avulla on myös mahdollisuus uuden resurssin luomiseen, jolloin se antaa tyhjän näytön. Resurssi ja sen attribuutit tallennetaan ”Save” - nappulaa painettaessa.

6. Muut Komponentit

Luvussa 6 pääkomponentit on esitelty osakomponentteina kuvausten avulla.

6.1 Tietosisältöluokat

6.1.1 DogmaResource

Luokka DogmaResource kapseloi resurssin tietosisällön.

Attribuutit tallennetaan resurssin sisällä HashMap<String, Object> -oliona, joka siis kuvaa attribuuttinimen arvolleen. Vaikka attribuuttiarvot tallennetaan Object-muodossa, on niiden tyyppi (vastaava java-luokka) koko ajan tiedossa, koska erityisessä Attribute-apuluokassa on globaali kuvaus attribuuttinimeltä sen tyypille, ja toisaalta DogmaResource:lle ei voi asettaa e.m. kuvauksessa määrittelemätöntä attribuuttia. Attribuuttiarvojen käsittelyyn on DogmaResource-luokassa useita apumetodeja; esimerkiksi käyttöliittymässä usein tarvittavia attribuuttiarvojen String-muunnoksia varten on omat metodinsa. Tällä hetkellä mahdollisia tyyppisiä attribuuteille ovat kokonaisluku, desimaaliluku, päivämäärä sekä teksti.

Tagit, jotka ilmentävät resurssin kuulumista osajoukkoihin, tallennetaan HashSet<String>-muodossa. Viitteet tallennetaan Vector<DogmaConnection>-muodossa, ja urit Vector<DogmaURI>-muodossa.

6.1.2 DogmaConnection

Luokka kuvaa viitettä kahden resurssin välillä. Näistä resursseista tiedetään aina vähintään niiden kanonisoidut nimet ja tyypit, mutta tilanteesta riippuen niihin voi olla myös olio (DogmaResource)-viite (viitettä ei aina ole, koska muuten koko kyseisen yhtenäisen komponentin viiteverkossa pitäisi olla java-olioina). Yhteys on aina suunnattu.

6.1.3 DogmaURI

Luokka on ohjelman käyttämä uri-laajennos. DogmaURI tallettaa sijaintitietoja (esim. www-osoite) ja muita attribuuttitietoja, kuten luontiajankohdan.

6.2 JDBC-rajapinta

JDBC-rajapinnan avulla voidaan luoda yhteys tietokantaan, suorittaa SQL-kyselyitä kantaan sekä muuttaa kannan sisältöä. JDBC-rajapinnassa ei ole tarkistuksia tai poikkeuksien sieppausta ja sitä suositellaan käytettävän ResourceBase-komponentin kautta. JDBC-rajapintaa toteuttaa ResourceBase-komponentin luokka BaseConnection.

6.3 Reader

Reader – komponentit ovat komponentteja erilaisten tiedostojen ja syötteiden metadatan lukemiseen. Jokainen Reader - komponentti lukee omanlaista dataansa riippuen toteutuksesta. Esimerkiksi PDFReader lukisi PDF-dokumenttien metadataa.

Readerit pyrkivät luomaan lukemastaan aineistosta resurssin ja viitteet, jotka osoittavat muihin toimijoihin. PDFReaderin tapauksessa lukija voisi saada dokumentista luettua dokumentin luojaan, jolloin voitaisiin luoda viite dokumentista luojaan ja luojaista dokumenttiin.

Readereiden määrää ei ole rajoitettu, mutta niiden kaikkien tulee toimia yhteistyössä Updater

- komponentin kanssa, jolloin niiden tulee toteuttaa rajapinnat iResource, iConnection ja iUri. Päivittäjä - komponenttia voi käyttää funktioiden addResource(iResource r), addConnection(iConnection c) ja addURI(iURI u, iResource r) kautta.

6.4 Konfiguraatitiedosto

Konfiguraatitiedostoon tallennetaan ne tiedot, jotka ovat voimassa istunnosta toiseen. Tähän kuuluvat oletustietokannan nimi, oletuskäyttäjän nimi ja salasana, istunnon alussa luettavat hakemistot ja <TODO muuta konfiguraatitiedostoon tarvittavaa>

6.5 Loki

Loki-komponentti pitää kirjaa tehdyistä tapahtumista. Lokin kirjoitus tapahtuu tekstitiedostoon, josta tapahtumia voi myös myöhemmin tarkastella. Lokin toteutuksessa käytetään String-vektoria, johon tapahtumat tallennetaan muodossa:

Lisäysaika#resurssinCanoname#tapahtumanId#tapahtumaKuvaus

Tallennettavat tapahtumat ovat: resurssin esiintyminen hakutuloksessa, resurssin liittäminen osajoukkoon, resurssin avaaminen lomakenäkymään, resurssin avaaminen oletusohjelmalla, resurssin liittäminen verkkonäkymään, resurssin attribuuttien muuttaminen.

6.6 DogmaDBCommComponent

DogmaDBCommComponent on komponentti jolla varmistetaan muutosten tallentuminen tietokantaan. HSQLDB – tietokanta tallentaa muutokset vain tietokantaa suljettaessa, eikä tietokantayhteyttä haluta sulkea ja avata jokaisen muutoksen yhteydessä.

Komponentti tallentaa muutetut resurssit staattiseen vektoriin, joka tallennetaan tietokantaan ohjelmaa suljettaessa. Komponentti myös avaa tietokantayhteyden ResourceBase – komponentin avulla ja sulkee sen myös tallennusten jälkeen.

6.7 DogmaClipboard

DogmaClipboard on luokka, jonka avulla käytetään käyttöjärjestelmän omaa leikepöytää DogmaResource-tyyppisten olioiden siirtämiseen ohjelman välillä instanssista toiseen. DogmaClipboard käyttää apunaan DogmaResourceTransfer-luokkaa, joka kirjoittaa DogmaResourcen leikepöydälle.

DogmaClipboardissa määritellään copy- ja paste-toiminnot ja sinne olisi syytä lisätä mahdolliset uudet leikepöytätoiminnot.

6.8 Ulkoiset komponentit

6.8.1 GUESS

GUESS on Hewlett-Packardin verkontutkimisohjelmisto, joka on toteutettu Javalla ja Jythonilla. GUESS käyttää verkon tietojen säilyttämiseen joko muistia tai HSQLDB-tietokantaa.

6.8.2 HSQLDB

HSQLDB on Javalla toteutettu relaatiotietokanta, jota voidaan käyttää niin suoraan muistista, kantaa tiedostosta lukien tai MySQL:n ja Postgresin tyyliin tietokantapalvelimena.

Oletusarvoisesti Dogma lukee tietokannan tiedostosta, mutta myös muut vaihtoehdot ovat mahdollisia. Niiden, etenkin tietokantapalvelimen, toiminta ei kuulu Dogman toiminnallisuuteen.

7. Testaussuunnitelma

7.1 Testauksen vaiheet

Testaus jaetaan kolmeen vaiheeseen; yksikkötestaukseen, integrointitestaukseen ja järjestelmätestaukseen.

7.1.1 Yksikkötestaus

Yksikkötestauksessa varmistetaan jokaisen luokan toimivuus. Jokainen luokka testataan sen toiminnallisuuden ja rakenteen kannalta. Toiminnallisuutta testattaessa varmistetaan, että luokka toimii juuri niin, kuin sen on tarkoituksin toimia. Yksikkötestauksessa pyritään 100% haaraumakattavuuteen. Tällä tarkoitetaan sitä, että luokan jokainen metodi, rakenne, ehtolause ja niiden haarautumien sisällöt käydään läpi. Jokainen ryhmän jäsen suorittaa itse yksikkötestauksen toteuttamilleen luokille.

7.1.2 Integrointitestaus

Integrointitestauksella varmistetaan jokaisen luokan toimivuus sen käyttämien rajapintojen ja komponenttien suhteen. Mikäli ryhmän jäsen kohtaa integrointitestauksessa ongelmia, voi jäsen pohtia niitä yhdessä sen ryhmän jäsenen kanssa, joka on suunnitellut ja toteuttanut käytettävän rajapinnan.

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

Dogman runkona toimii resurssitietokanta. Muut osat ovat käytännössä hyödyttömiä ilman toimivaa tietokantaa. Resurssitietokantaan liitetään järjestyksessä muita osia, jotka toimivat sen avulla. Nämä osat ovat järjestyksessä: päivittäjä, käyttöliittymä, loki ja verkkonäkymä.

7.1.3 Järjestelmätestaus

Järjestelmätestauksella varmistetaan toteutettavien toiminnallisten ja ei toiminnallisten toimintojen oikeellisuus. Myös käyttötapaukset ja ohjelmiston toiminto kohdekäyttöympäristössä testataan. Käyttötapauksena toimii asiakkaan luoma esimerkkikäyttötapaus:

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

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

mitä porukkaa on kyseessä.

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

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

7.2 Testattavat toiminnot

Ainoastaan toteutettavat toiminnalliset ja ei-toiminnalliset vaatimukset testataan. Totettavat toiminnot ovat:

- Verkkonäkymä
- Resurssitaulukko
 - Resurssitaulukon kentät
 - Resurssitaulukon hakeminen tietokannasta
 - Resurssitaulukon järjestäminen attribuuttien mukaisesti
 - Lomakenäkymän avaaminen resurssitaulukosta
- Resurssin tietojen muokkaus
 - Resurssin tietojen muokkaus lomakenäkymässä
 - Resurssin lisääminen
- URI-taulukko
 - URI:n lisäys
 - URI:n poisto
 - URI:n avaaminen
- Tapahtumien seuranta ja tellennus
- Resurssileikepöytä
 - Kopiointi
 - Liittäminen
- Tietokannan valinta
- Testausviitteiden luonti
- Toteutuskieli (englanti, ts. kielioppi kunnossa, ei kirjoitusvirheitä)

7.3 Testausympäristö

Kaikki testaukset suoritetaan tietojenkäsittelytieteen laitoksen linux-koneissa KDE-ympäristössä.

8. Liitteet