

# Käyttöliittymäsuunnitelma

MAITO – Metadatan Analyysi-, Integraatio- ja Transformaatio-Ohjelmisto

## 1. Johdanto

Jo määrittelyvaiheessa kävi ilmi hyvinkin tarkkaan minkälaisia asioita asiakas haluaisi tehdä ja "miten": sen sijaan, että suunnittelun olisi voinut aloittaa puhtaalta pöydältä, oli näitä asioita pakko angeta mukaan jonka ansiosta käyttötapaukset ovat pahimmillaan hyvinkin sidottuja järjestelmään - hyvien ja havainnollisten toiminnallisten käyttötapaustenkin keksiminenkin oli vaikeaa eikä täysin onnistunut kuten lopputuloksesta näkyy.

Tämän lisäksi aikataulun tiukkuuden takia mutkia on oitettu melkomoisesti milloin vain järkevästi mahdollista. Prototyyppi kasvoi maagisesti paperille lyijykynällä epäselvästi söhrätyistä käyttötapausten läpimenojen merkinnöistä puhtaaksi piirretyiksi staattisiksi näyttökuviksi joissa on yritetty ottaa valmiiksi huomioon teknisiä rajoitteita joita projektissa tulee vastaan (esimerkiksi jatkuvaa hakua tuskin saadaan toteutettua gigojen kokoisille tietokanta-aineistoille). Käyttötapauksen läpimenojen kuvasarjoja ei ehditty tehdä - näissä staattisissa kuvissa on muutama kommentti punaisella toiminnallisuudesta.

Itse käyttöliittymäprototyyppi on liitteenä erillisessä tiedostossa – tässä dokumentissa on kuvailtuna rajapinnat ja käyttötapaukset.

## 2. Datankeruun ja integroinnin rajapintavaatimukset käyttöliittymälle

### 2.1. Yhteinen statusrajapintavaatimus

- Molemmille komponenteille tarvitaan jonkinlainen statuksenpalautuskeino jotta käyttöliittymä voi estää käyttäjältä tietyt samanaikaiset toiminnot ja toisaalta näyttää käyttäjälle tietoa siitä, mitä ohjelma milloinkin tekee
- Molempien komponenttien statuksenpalautusrajapintojen pitäisi olla samanlaiset jotta niitä voitaisiin käsitellä helposti samalla tavalla
- Minimissään tarvitaan tieto siitä onko komponentti idlenä vai tekeekö jotain - tähän riittäisi vaikka boolean-arvo
- Mielellään mahdollisen tarkkaa tekstuaalista tietoa siitä mitä komponentti tekee missäkin vaiheessa: esim. "idle + viimeisin suoritettu operaatio", "hakee dataa lähteestä x", "integroi/transformoi lähteen x dataa", "poistaa lähdetä x", ym. - tällöin voidaan näyttää käyttäjäystävällistä statustietoa ja sen lisäksi tietää onko komponentin toiminta käynnissä vai ei
  - Tällöin palautettavasta tiedosta tulee voida helposti havaita onko kyseessä statuksen muuttuminen operaatiosta toiseen vai operaatiosta idleksi (esimerkiksi: kun komponentti on lopettanut tehtävänsä, palautettavan merkkijonon ensimmäiset merkit ovat "idle" jolloin käyttöliittymä tietää että voi sallia aiemmin lukitut operaatiot eikä kyseinen komponentti tee enää mitään ennen seuraavaa sille annettua käskyä)
- Paras olisi toki tämän lisäksi vielä operaatioiden arvioitu kesto jolloin voitaisiin näyttää jonkinlainen progress-palkki

- Mahdollisia toteutustapoja:
  - Käyttöliittymäkomponentti kutsuu tasaisin väliajoin transformointi/integraatio-hallintakomponentin metodia kun komponentti ollaan käynnistetty, metodi palauttaa tiedon
  - Komponentilta saadaan operaation käynnistäessä (tai muuten halutessa) viite jaettuun olioon / muuttuun josta voidaan lukea status tasaisin väliajoin
  - Jos keksitte paremman vaihtoehdon, käyttäkää sitä - nämä ovat vain ehdotuksia ja niissä voi olla teknisiä vaikeuksia / esteitä

## 2.2. Datankeruun rajapintavaatimukset

- Lähteiden listaus:
  - Ei kutsuparametreja
  - Palauttaa joukon/listan jossa kustakin lähteestä vähintään lähteen nimi (ja yksikäsitteinen ID jos nimi ei yksilöi), lähteen koko, päivytyspäivämäärä (ja muutospäivämäärä), lähdetyyppi ja formaatti, integrointistatus (ts. onko integroitu jo)
- Lähteen poisto:
  - Parametrina joukko/lista lähteen nimiä tai muita yksikäsitteisiä ID:itä 1-n kappaletta
  - Käynnistää datankeruuoskomponentissa annettujen lähteiden poiston
  - Ei paluuarvoa, poiston yhteydessä käyttöliittymä kysyy statusrajapinnalta tilanteesta
  - Poiston jälkeen käyttöliittymä kutsuu lähteiden listausta uudestaan jotta näkyvä lista saadaan päivitettyä
- Lähteen päivitys:
  - Parametrina joukko/lista lähteen nimiä tai muita yksikäsitteisiä ID:itä 1-n kappaletta
  - Käynnistää datankeruuoskomponentissa annettujen lähteiden päivityksen
  - Ei paluuarvoa, päivityksen ollessa käynnissä käyttöliittymä kysyy statusrajapinnan kautta tilanteesta
  - Päivityksen jälkeen käyttöliittymä kutsuu lähteiden listausta uudestaan jotta näkyvä lista saadaan päivitettyä
- Lähteen lisäys:
  - Parametrit: nimi, lähdetyyppi (tiedosto / OAI-PMH), lähteen URL tai polku (tosin tiedostopolku voidaan ilmaista myös file:// -urlina...), formaatti (pikaformaatti / DCXML / oai\_citeseer)
  - Käynnistää datankeruuoskomponentissa määritellyn lähteen lisäyksen, noudon ja transformoinnin
  - Ei paluuarvoa: datan nouto ja transformointi käynnistyy ja aletaan kyselymään statusrajapinnan kautta tilanteesta
  - Lisäyksen jälkeen käyttöliittymä kutsuu lähteiden listausta uudestaan jotta näkyvä lista saadaan päivitettyä

### 2.3. Integroinnin rajapintavaatimukset

- Resurssiverkkojen listaus
  - Ei parametreja
  - Palauttaa joukon/listan, jossa kunkin resurssiverkon nimi (ja yksikäsitteinen ID jos nimi ei yksilöi)
- Resurssiverkon sisällön listaus
  - Parametrina yhden resurssiverkon nimi / muu listauksessa saatu yksikäsitteinen ID
  - Palauttaa joukon/listan jossa valitussa resurssiverkossa olevien lähteiden nimet ja integrointipäivämäärät (milloin mikäkin lähde viimeksi integroitu tähän resurssiverkkoon)
- Resurssiverkon lisäys:
  - Parametrina uuden resurssiverkon nimi merkkijonona
  - Käynnistää integrointikomponentissa uuden resurssiverkon luomisen
  - Ei paluuarvoa, luomisen ollessa käynnissä käyttöliittymä kyselee statusrajapinnan kautta tilanteesta
  - Lisäyksen jälkeen käyttöliittymä kutsuu resurssiverkkojen listausta uudestaan
- Lähteen integrointi:
  - Parametrit: joukko/lista lähteen nimiä tai muita datankeruun lähteiden listauksen palauttamia yksikäsitteisiä ID:itä 1-n kappaletta, yhden resurssiverkon nimi tai sen muu listauksessa saatu yksikäsitteinen ID
  - Käynnistää integrointikomponentissa integrointiprosessin: integroi valitut lähteet valittuun resurssiverkkoon
  - Ei paluuarvoa, integroinnin ollessa käynnissä käyttöliittymä kyselee statusrajapinnan kautta tilanteesta
  - Integroinnin jälkeen käyttöliittymä kutsuu resurssiverkon sisällön listausta ja datankeruun lähteiden listausta uudestaan
- Lokitietojen tarkastelu:
  - Parametrina yhden resurssiverkon nimi / muu listauksessa saatu yksikäsitteinen ID
  - Palauttaa merkkijonona resurssiverkkoon liittyvät integroinnin lokitiedot: tätä varten pitää integroinnin lokitiedot tallettaa joko tiedostoon tai tietokantaan automaattisesti integraatioprosessin loppuksi (tämä voidaan tehdä myös muulla tavalla jos kuulostaa huonolta ratkaisulta)

### 2.4. Haun ja selauksen rajapintavaatimukset

Kun ensimmäisessä käyttöliittymäruudussa on valittu resurssiverkko ja painettu "avaa verkko tarkasteluun"-nappia, siirrytään hakuun ja selaukseen (käyttöliittymäruudut 2 ja 3). Haku ja selaus toteutetaan kiinteästi yhteen varsinaisen käyttöliittymän kanssa joten erillisiä vaatimuksia arkkitehtuuria varten ei tarvita.

### 3. Käyttötapaukset

Näitä skenaarioita on käytetty pohjana käyttöliittymän suunnittelulle jonka jälkeen vaatimukset on vielä kerran käyty läpi ja varmistettu että toteutettavat vaatimukset pystytään tekemään tällä käyttöliittymällä - tässä vaiheessa lisäilyjä käyttöliittymään tuli vielä paljon.

#### 3.1. Käyttötapaus 1: Merkityksiä tietoon ja talteen

Maanantaina 31.10.2005 klo 10 Matti on yliopistolla koneensa ääressä työskentelemässä. Hän on juuri saanut idean siitä, miten erilaisten resurssien hyvyttä ja luotettavuutta voisi merkitä erityyppisin resurssien välisin yhteyksin. Matilla on käytössään ohjelma jolla hän voi analysoida verkkoja tarkemmin, mutta hänellä ei ole sopivia lähdetiedostoja analysointiin ja niiden käsin tyhjästä tekeminen vaatisi liikaa työtä - helpoin vaihtoehto olisi käyttää olemassaolevaa pohja-aineistoa ja lisätä merkityksiä päälle käsin.

##### 3.1.1. Tilatiedot

- Matin käyttämä analysointiohjelma lukee Pajek-listatiedostoja
- Matti tarvitsee analysointiohjelmaa varten verkon sekä itse resursseista että merkityksistä
- Matilla on kovalevyllään citeseerin tietokanta oai\_citeseer-muodossa tiedostona sekä dokumentti- ja nimityyppiset itse tekemänsä pikaformaattitiedostot - hän tietää että näistä yhdistelemällä saisi hyvää metadataa idean tutkimiseen
- Matin esimerkkiaineistoon analyysia varten lisättävät merkitykset:
  - Jokin toimija: luotettavuus, painoarvo 5
  - Jokin toimija -> jokin toinen toimija: suosittelee, painoarvo 4
- Matti haluaa että pohjamateriaali ja siihen tehdyt muutokset ovat helposti käytettävissä myöhemminkin
- Matti tietää että kaikissa lähteissä on myös turhaa tietoa fysiikan artikkeleista joita ei pitäisi tulla ollenkaan mukaan lopulliseen analysoitavaan tietoon
- Matti on kiinnostunut tässä analyysissa vain MIT Pressin julkaisemista artikkeleista
- Mattia kiinnostaa tietää kuinka paljon resursseja pohja-aineistoissa on ja tietää tarvitsevänsä tätä tietoa analyysin myöhemmissä vaiheissa
- Mattia kiinnostaa myös nähdä kokonaiskuvana miten resurssit ovat liittyneet toisiinsa niin, että hän voi jotenkin tunnistaa mikä resurssi on liittynyt mihinkin

#### 3.2. Käyttötapaus 2: Toinen idea

Maanantaina 31.10.2005 klo 18 tutkittuaan ensimmäisen käyttötapauksen synnyttämää materiaalia Matti keksii toisen vaihtoehtoisen hyvän idean, mutta ensimmäisen idean tutkimisessa syntynyt materiaali ei ole sopivaa.

### 3.2.1. Tilatiedot

- Matilla on käytössä sama ohjelma kuin tapauksessa 1
- Matilla on tiedossa kaksi OAI-PMH-lähdettä jotka sisältävät analysointiin hyvää metadataa - toisesta lähteestä on jo olemassa paikallinen kopio mutta se on haettu kaksi kuukautta sitten ja lähteessä voi olla uuttakin tietoa joka pitää saada mukaan. Matti arvelee, että näistä lähteistä yhdistelemällä saisi hyvää metadataa toisen idean tutkimiseen.
- Matti ei ole kiinnostunut pohja-aineiston resursseista joista ei ole yhteyksiä toisiin resursseihin
- Matti tietää, että toisessa lähteistä on Antti Niemisen artikkeli "Managing risks in software engineering projects" jonka julkaisija on oikeasti MIT Press - virheellisesti Microsoft Press - jonka julkaisijatiedot pitää korjata analyysia varten

### 3.3. Käyttötapaus 3: Monimutkaisempi haku

Tiistaina 1.11. klo 14 Matti on miettinyt koko yön ensimmäistä ideaa ja toteaa, että hieman erilaisen otannan analysointi samasta materiaalista olisikin paikallaan. Tällä kertaa Mattia kiinnostaa saada suoraan tietynlainen otanta talteen toista analysointiohjelmaa varten

#### 3.3.1. Tilatiedot

- Matin on helpompi hahmottaa haluamansa monimutkaisemmat hakuehdot jonkinlaisena rakenteisena kyselykielenä, esimerkiksi SQL-lauseena, kuin täyttelemällä yksittäisiä kenttiä
- Matin käyttämä toinen analysointiohjelma lukee CSV-matriisimuotoisia tiedostoja
- Matti haluaa käyttää aiemmin tapauksessa 1 tekemäänsä resurssiverkkoa uuden otannan kanssa vaikkei otannassa olisikaan kaikkia ensimmäisen tapauksen resursseja