MELA2.0 – Käyttäjäkuvaukset



Luonnonvarakeskus

05.08.2019

LUONNOS v0.3

# Sisällys

[Sisällys 2](#_Toc16061312)

[1. Dokumentin tarkoitus 1](#_Toc16061313)

[2. MELA2.0-metsätietojärjestelmän rajaus 2](#_Toc16061314)

[3. Arkkitehtuurikuvaus 3](#_Toc16061315)

[3.1. Asiakas-palvelin arkkitehtuuri MELA2.0:ssa 3](#_Toc16061316)

[3.2. Järjestelmän yleiskuvaus 4](#_Toc16061317)

[3.3. Komponentit 5](#_Toc16061318)

[4. Järjestelmän käytönkuvaus 7](#_Toc16061319)

[4.1. Käyttäjät 7](#_Toc16061320)

[4.2. Käyttäjätarinat 7](#_Toc16061321)

[4.3. Käyttötapaukset 9](#_Toc16061322)

# Dokumentin tarkoitus

Tässä dokumentissa on kuvattu MENU-projektiin liittyvän MELA2.0-metsätietojärjestelmän keskeiset vaatimukset ja käyttötapaukset. Dokumentti on tarkoitettu seuraaville kohderyhmille:

* **Metsätieteilijät**, joilla on tieto siitä mitä kuvatulla metsätietojärjestelmällä pitää pystyä tekemään.
* **Sovelluksen kehittäjät**, joiden tulee ymmärtää ne toiminnalliset vaatimukset, jotka tietojärjestelmään tulee sisällyttää.

Tämä dokumentti toimii keskeisenä rajapintana, sille kommunikaatiolle, joka vaaditaan järjestelmän toteuttamista varten. Dokumentin tarkoitus on yhdenmukaistaa ajatukset MELA2.0-metsätietojärjestelmän käyttäjistä ja vaatimuksista.

Tämän dokumentin ovat laatineet tutkimusprofessori Tuula Packalen ja erikoistutkija Jari Vauhkonen sekä tutkijat Reetta Lempinen ja Daniel Melander. He myös vastaavat dokumentin ylläpidosta.

# MELA2.0-metsätietojärjestelmän rajaus

MELA2.0 on Luonnonvarakeskuksen suunnittelema ja kehittämä tietojärjestelmä, jonka avulla on mahdollista simuloida ja ennustaa kasvu- ja kehittymisskenaarioita metsävara-aineistolle. Järjestelmän vaatimuksia on muun muassa:

* Suuralueiden skenaariolaskelmat
* Hakkuumahdollisuusarviot
* Erilaisten metsänhoitostrategioiden vaikutusarviot
* Maanomistajien omien metsäresurssien optimointi

MELA2.0-metsätietojärjestelmä on tarkoitus syrjäyttää vanha MELA-tietojärjestelmä ja tuomaan mukanaan läpinäkyvyyttä sekä avoimuutta metsäresurssien hyödyntämiseen, niin yksityisille metsänomistajille kuin poliittisille päätöksentekijöille. Lisäksi kuka tahansa voi osallistua järjestelmän kehittämiseen, sillä järjestelmän lähdekoodi on kaikille avoin.

Järjestelmästä hyötyvät niin yksityinen kuin julkinen sektori. Järjestelmä tarjoaa tukea mm.:

* Luonnonvarakeskuksen viranomaistehtävissä.
* Kansallisessa ja kansainvälisessä raportoinnissa.
* Erilaisissa tilasto- ja raportointipalveluissa.
* Tutkimushankkeissa.
* Ministeriöiden asiantuntijapalveluiden laskelmissa.
* Metsä- ja energiateollisuuden investointilaskelmissa.
* Metsäresurssien kehityksen ennustamisessa, vaikutusarvioissa ja optimoinnissa.
* Metsäyhtiöiden ja Metsähallituksen vaihtoehto- ja hakkuulaskelmissa.

Tietojärjestelmän perusarkkitehtuuri on asiakas-palvelin-arkkitehtuuri, mutta siinä on viitteitä myös kerrosarkkitehtuurista ja tietovuoarkkitehtuurista etenkin järjestelmän komponenttien osalta. Järjestelmän käyttö tapahtuu erillisen asiakassovelluksen kautta, jonka käyttäjä voi asentaa omalle paikalliselle tietokoneelle MELA:n verkkosivuilta. Asiakassovellus tarjoaa käyttöliittymän, jonka avulla käyttäjä tarjoaa syötetietoja varinaiselle järjestelmälle. Varsinainen järjestelmä ja sen loogiset komponentit sijaitsevat Valtorin palvelimilla. Myöhemmässä vaiheessa asiakassovelluksesta on tarkoitus kehittää myös riisuttu selainpohjainen versio.

Erillisen asiakassovelluksen lisäksi Järjestelmä koostuu yhteensovitetuista komponenteista, jotka tarjoavat sovellusrajapintoja hyödyntäen saumattoman laskennan metsikködatalle. Kriittisimpiä komponenttikokonaisuuksia ovat mm. Hakukone, Simulaattori ja Optimoija (komponenteista tarkemmin luvussa 3). Komponenttien toiminnalliset vaatimukset sekä arkkitehtuuri on kuvattu erillisillä dokumenteilla.

Laskennassa käytettävä metsätieto on sähköisessä muodossa olevaa informaatiota, jonka rakenne voi vaihdella rasterimuotoisesta hiladatasta perinteiseen maastosta kerättyyn informaatioon. Selkeitä standardoituja data-aineistoja ovat VMI ja SMK. Järjestelmän ominaisuuksiin kuuluu annettujen tietojen esiprosessoimaan ja soveltaminen, siten että laskennassa saavutettaisiin optimaalinen tulos.

MELA2.0 on edistyksellinen metsätietojärjestelmä, joka sisältää useita erilaisia laskennallisen älykkyyden ratkaisuja. Metsikködatan täydentäminen esiprosessoimalla, mallifunktioiden etsiminen ja hyödyntäminen ilman käyttäjän eksplisiittistä kuvausta ja optimointivaihtoehtojen automaattinen karsiminen kuuluvat jokaiseen yksittäiseen simulaatiotapaukseen.

Järjestelmä tuottaa raportteja perustuen syötetietona tarjottuun metsikködataan, tuloskuvauksen ja simuloinnin ohjausskriptiin. Raportit ovat optimoituja tuloksia laskentaskenaarioista tietyltä aikaväliltä. Formaatin voi itse valita.

# Arkkitehtuurikuvaus

## Asiakas-palvelin arkkitehtuuri MELA2.0:ssa

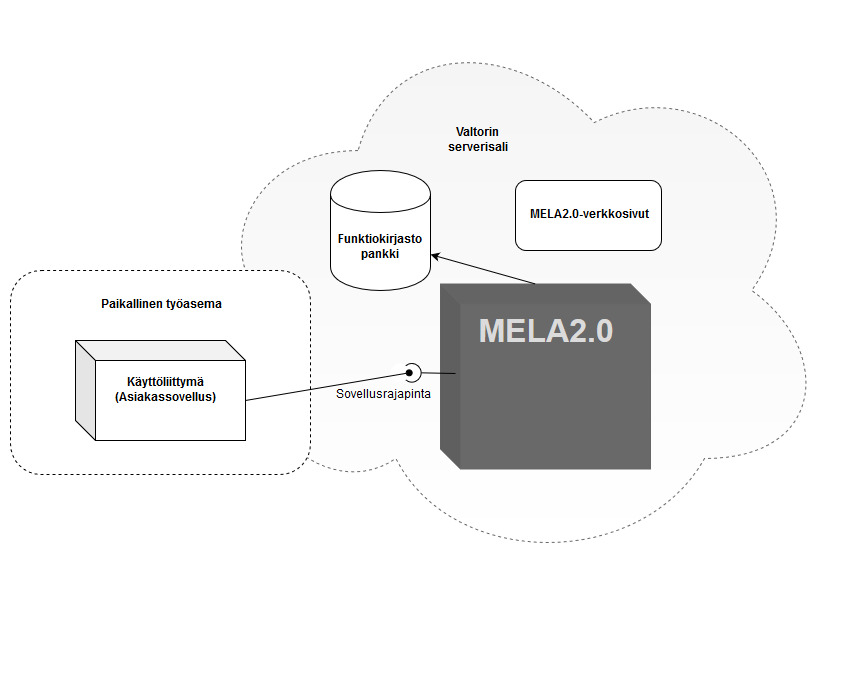
Asiakas-palvelin-arkkitehtuurissa asiakkaan pyyntö käsitellään palvelimella. AP-arkkitehtuurissa sovellukset on hajautettu laskennallisiin yksiköihin. Kommunikaatio asiaakaan ja palvelimen välillä tapahtuu verkkoyhteydessä. Arkkitehtuurissa palvelin kuvastaa abstraktio metsikköinformaatiolle. Asiakassovelluksen käyttäjän ei tarvitse tuntea palvelimen toimintaa pystyäkseen käyttämään järjestelmää. Viestien välittäminen asiakkaan ja palvelimen välillä tapahtuu pyyntö-vastaus viestintä protokollaa hyödyntäen.

Usein palvelin ympäristöön toteuttaa myös sovellusrajapinnaksi kutsuttu abstraktio, joka vastaa välitettävän tiedon oikeasta standardoimisesta. Sovellusrajapinnalla on tarkoitus säännöstellä syötetietona annetun datan formaattia, mutta lisäksi mahdollistaa datan oikeellisuus laskentaa varten. Toisin sanoen, sovellusrajapinta yhdenmukaistaa asiakassovelluksen ja palvelimen välisen datan.

MELA2.0-järjestelmän ja asiakassovelluksen toiminta kuvataan AP-arkkitehtuurin mukaisesti. MELA2.0-kokonaisarkkitehtuurissa asiakassovellus on sijoitettu loppukäyttäjän paikalliselle levyasemalle. Asiakassovelluksen käyttöliittymällä käyttäjän on mahdollista luoda kuvaus halutusta laskennasta tietylle metsikködatalle. Asiakassovelluksen kautta käyttäjän laskennallinen toimeksianto välitetään sovellusrajapinnan kautta palvelimelle, jossa sijaitse varsinainen MELA2.0-järjestelmä ja sen simulointiyksikkö. MELA2.0-palvelin suorittaa laskennan ja palauttaa asiakkaalle raportin laskennan tuloksesta. Raportti on tarkasteltavissa käyttöliittymän välityksellä.

## Järjestelmän yleiskuvaus

Järjestelmän käyttöliittymänä toimii itsenäinen (stand-alone) asiakassovellus. Varsinainen MELA2.0-järjestelmä on sijoitettu Valtorin-palvelimelle, joten simulointitoimeksiantojen välittäminen tapahtuu asiakassovelluksella. Valtorin palvelimille on sijoitettu lisäksi tietokanta sertifioiduille mallipaketeille (nk. funktiokirjasto) sekä MELA2.0-järjestelmän viralliset kotisivut. Kuvassa 1 on havainnollistettu järjestelmän sijoittelua.



Kuva 1: Yleiskuvaus MELA2.0-järjestelmästä.

## Komponentit

Tässä luvussa on kuvattu MELA2.0-järjestelmän keskeisimmät komponentit ja niiden tarkoitus yleisellä tasolla. Tarkemmat kuvaukset komponenteista löytyvät komponenttikohtaisista vaatimusmäärittelydokumenteista. Komponenttien yhteenveto on nähtävissä kuvasta 2.

**Käyttöliittymä** – Loppukäyttäjän käyttöliittymä (asiakassovellus), jossa voi valita laskennassa käytettävän datan, muodostaa tuloskuvauksen, esitellä hyödynnettävät mallipaketit sekä valita simulointiskriptejä.

**Funktiokirjasto** – Koostuu sertifioiduista mallipaketeista. Sertifioitu mallipaketti koostuu MELA2.0-standardin mukaisesti kirjoitetuista mallifunktioista. Mallifunktio on standardoitu, kun funktio on toteutettu MELA2.0-tyylin mukaisesti, funktioiden metadata on kuvattu mallien kommenteissa ja funktion toimivuus osana MELA2.0-järjestelmää on validoitu. Mallipakettien ensisijainen ohjelmointikieli on R-kieli.

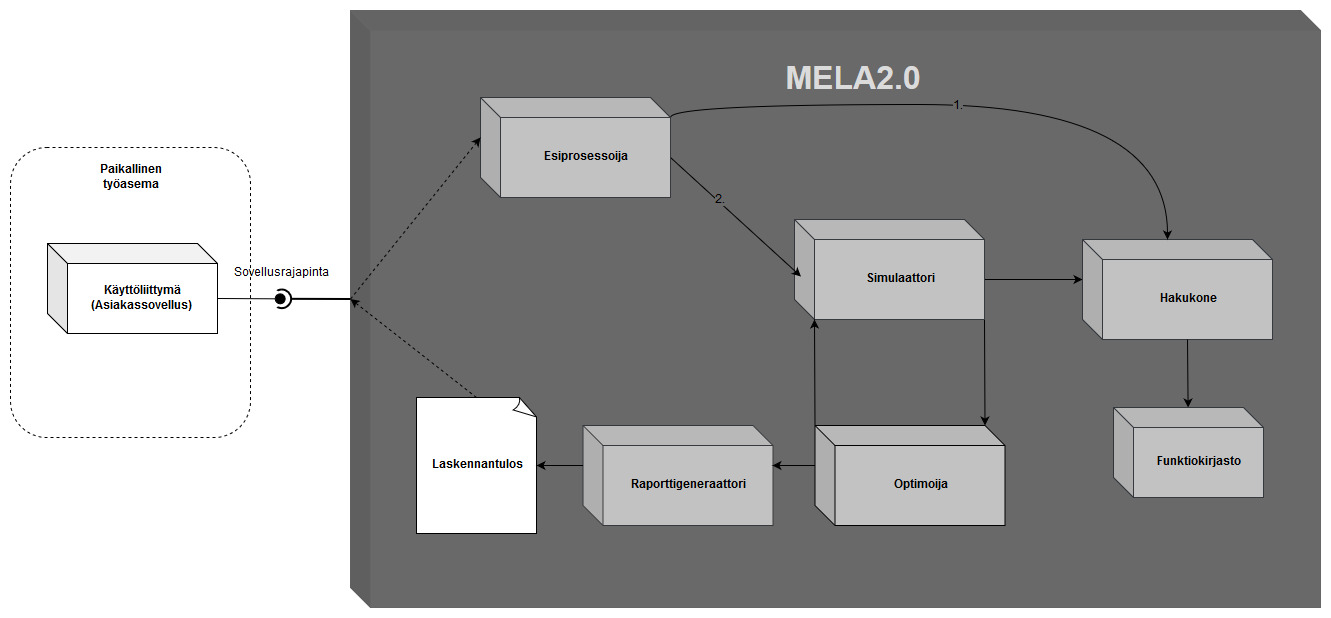
**Esiprosessoija** – Esiprosessoinnin tarkoitus on saattaa syötedata sellaiseen muotoon, että simulointi on mahdollista. Esimerkiksi, jos annetusta datasta puuttuu olennaisia muuttujia, esiprosessointi pyrkii täydentämään puuttuvat muuttujat dataan. Täydennysprosessi hyödynnetää hakukonetta, joka annettua syötedataa tulkitsemalla tunnistaa ne funktiokirjaston mallifunktiot, joilla täydennys onnistuu.

**Hakukone** – Hakukone vastaa funktiokirjaston mallipaketeissa olevien funktioiden tulkitsemisesta. Hakukone lukee mallifunktioiden metadataa, jonka perusteella se pystyy sijoittamaan funktiot sopivaan kutsujärjestykseen perustuen syötetietona annetulle datalle ja tuloskuvaukselle. Näin syntyy laskentaketjuja, joita voidaan hyödyntää syötedatasta puuttuvan tiedon laskemisessa tai simuloinnin toteuttamisessa. Laskentaketjut muodostetaan funktiokirjaston mallifunktioista.

**Simulaattori** – Laskentaketjujen suorittaminen.

**Optimoija** – Vaihtoehtojen karsiminen.

**Raporttigeneraattori** – Raporttien tulostaminen.



Kuva 2: MELA2.0-järjestelmän keskeisimmät komponentit sekä komponenttien välinen kommunikaatio.

# Järjestelmän käytönkuvaus

## Käyttäjät

Käyttäjiksi on tunnistettu sellaiset sidosryhmänjäsenet, jotka ovat suorassa vuorovaikutuksessa järjestelmän kanssa asiakassovelluksen kautta. Käyttäjäryhmien vuorovaikutus kohdistuu toisistaan poikkeaville komponenteille MELA2.0-järjestelmässä. Käyttäjät on eritelty neljään pääryhmään: loppukäyttäjät, mallintajat, kirjastonhoitajat ja MelaXpertit. Yksittäinen käyttäjä voidaan tunnistaa edustavan useampaa ryhmää. Tietyt käyttäjäryhmät sisältävät alaryhmiä.

Tietojärjestelmän käyttäjäryhmien erittely:

**Loppukäyttäjä** – Järjestelmän loppukäyttäjän päätarkoituksena on simuloida metsikködatalle vaihtoehtoraportteja. Loppukäyttäjän ja MELA2.0-järjestelmän välinen kommunikaatio tapahtuu asiakassovelluksen välityksellä. Loppukäyttäjät jakautuvat alaryhmiin, joita ovat metsänomistajat, yritykset, ministeriöt, tutkijat ja testaajat.

**Mallintaja** – Mallintajat ovat järjestelmän sisällöntuottajia. Heidän pääasiallinen tarkoituksensa on mallipakettien tuottaminen osaksi funktiokirjasto komponenttia. Sisällöntuottaminen voi olla paikallista tai julkista. Julkinen mallipaketti sertifioida osaksi MELA2.0-järjestelmää, jolloin mallintaja osallistuu funktiokirjaston laajentamiseen avoimen lähdekoodin lisenssien mukaisesti.

Sisällöntuottaminen tapahtuu julkaisemalla mallintajan laatimia laskentamallintaja valitulla ohjelmointikielellä ja lisäämällä malleihin MELA2.0:n käyttöä ohjaavat metadata kuvaukset (nk. metatagit). Koska mallintajan tehtävä vaatii syvällisempää ymmärrystä järjestelmän toiminnasta, pitää käyttäjäryhmälle tarjota opiskelumateriaalia sekä koulutustilaisuuksia järjestelmän käytöstä ja ajantasaista informaatiota järjestelmän uusista ominaisuuksista.

**Kirjastonhoitaja** – Vastaa mallipakettien sertifioinnista ja julkaisemisesta avoimen lähdekoodin lisenssien mukaisesti. Kirjastonhoitaja vastaanottaa mallipaketteja mallintajalta, jonka jälkeen hän validoi, testaa ja lisää tarvittavat metakuvaukset paketin mallifunktioihin. Sertifioitu mallipaketti liitetään osaksi funktiokirjastokomponenttia, jolloin mallipaketti myös julkaistaan avoimena lähdekoodina.

**MelaXpertti** – Xpertti vastaa jo sertifioitujen funktioiden malliparametrien sekä virhekertoimen kalibroinnista. Xpertin tehtäviin kuuluu lisäksi laatia simulointiskriptejä.

**Avoin** **kehittäjä** – Avoimen lähdekoodin kehittäjät osallistuvat järjestelmän toiminnallisuuksien parantamiseen, toteuttamiseen ja ylläpitoon.

## Käyttäjätarinat

**Mallipakettien julkaiseminen** –Minna on tutkija, joka on kehittänyt puutason kasvumalleja. Hän haluaa hyödyntää kehittämiään malleja MELA2.0-simulaattorissa. Minna suuntaa MELA2.0-verkkosivuille tutustumaan miten julkaisuprosessi tapahtuu. Luettuaan ohjeet Minna päättää toteuttaa funktiopakettinsa R-kielellä, koska se on MELA2.0:n ensisijainen ohjelmointikieli mallifunktioille. Minna toteuttaa mallipakettinsa MELA2.0-standardien mukaisesti R-studiolla ja lisää lopuksi dokumentaatiossa kuvatut metatagit funktioiden kommenteiksi.

Luetun dokumentaation perusteella, Minna osaa ottaa yhteyttä kirjastonhoitajaan, joka vastaa uusien mallipakettien yhteensovittamisesta osaksi MELA2.0-funktiokirjastoa. Kirjastonhoitaja vastaa Minnalle ja esittää kolme vaihtoehtoa. Joko Minna julkaisee pakettinsa avoimesti julkaisualustalla (GitHub yms.) tai lähettää mallipaketin joko sähköpostilla tai verkkosivujen kautta kirjastonhoitajalle. Kirjastonhoitaja mainitsee vielä, että verkkosivujen kautta tapahtuva julkaiseminen on suositeltavaa.

Minna ei ole aikaisemmin julkaissut mitään kolmannen osapuolen julkaisualustoilla, joten hän päättää suunnata MELA2.0-verkkosivuille ja lähettää mallipaketin verkkopalvelun kautta. Minna täyttää julkaisulomakkeen, jossa pyydetään hänen yhteystietojaan ja lisenssilupaa julkaista modifioitu mallipaketti avoimena lähdekoodina. Parin viikon kuluttua Minna saa sähköpostia, jossa ilmoitetaan, että mallipaketti on sertifioitu MELA2.0 yhteensopivaksi. Nyt Minna voi hyödyntää omia mallejaan osana MELA2.0-järjestelmää.

**Yksinkertainen simulointi tapaus** – Lauralla on hallussaan metsikködataa XML-muodossa. Hän haluaa simuloida MELA2.0:lla metsikön kasvuennustetta 20 vuoden päähän. Laura käynnistää MELA2.0:n asiakassovelluksen tietokoneeltaan ja valitsee käyttöliittymän avulla XML-tiedoston, joka sisältää laskennassa käytettävän metsikködatan. Seuraavaksi Laura muodostaa tuloskuvauksen, siitä millaisen raportin hän haluaa simuloinnin lopputuloksena. Hän valitsee tuotettavaksi raportin, joka maksimoi metsikönkasvun seuraavalle 20 vuodelle.

Seuraavaksi Laura valitsee hänen tarpeelleen sopivamman simulointiskripti. Hän selaa erilaisia vaihtoehtoja lukemalla skriptien yhteenvetoja ja käyttötapauksia MELA2.0-verkkosivulta. Hän valitsee skriptin, joka vastaa parhaiten hänen haluamaansa lopputulosta ja määrittelee skriptin käyttöönoton MELA2.0:n käyttöliittymässä. Lopuksi Laura tarkastaa yhteenvedon hänen asettamistaan syöteparametreista (data, tuloskuvaus ja simulointiskripti) ja käynnistää simuloinnin. Simuloinnin päätteeksi MELA2.0 tuottaa raportin, joka sisältää Lauran määrittelemän laskennan lopputuloksen. Laura tallentaa raportin .csv tiedostona haluamaansa hakemistoon tietokoneelleen.

**Mallipaketin yhteensovittaminen osaksi funktiokirjastoa** – Kalle toimii kirjastonhoitajana MELA2.0-järjestelmälle. Eräänä päivänä hän vastaanottaa R-kielellä kirjoitetun kasvumallipaketin sähköpostilla mallintajalta. Paketti on .zip muodossa, joten Kalle purkaa paketin omalle tietokoneelleen ja tutkii paketin sisältöä. Hän toteaa, että paketti on R-standardien mukaan kirjoitettu ja sisältää asiaan kuuluvat metatagit.

Kalle käy läpi paketissa olevat funktiot yksi kerrallaan ja lisää kirjastonhoitajan metatagit funktioiden kommentteihin. Samalla hän määrittelee mallifunktioiden arvoalueet, jolla funktioiden käyttö on mahdollista/optimaalista. Lisäksi Kalle käy läpi funktioiden toimintaa ja validoi, että ne toimivat suunnitellusti ja antavat järkeviä tuloksia esimerkkidatalle. Lopuksi Kalle vielä testaa, että hakukone pystyy löytämään kaikki funktiota ja luomaan niistä sopivia laskentaketjuja.

Kun Kalle on saanut tehtyä edellä mainitut työvaiheet hän sertifioi mallipaketin ja lisää sen osaksi funktiokirjastoa. Samalla hän julkaisee mallipaketin julkaisualustalla (GitHub yms). Viimeisenä hän informoi mallintajaa siitä, että julkaisuprosessi on nyt valmis ja mallipaketti on saanut MELA2.0-sertifikaatin.

**Funktiokirjaston mallifunktioiden kalibroiminen** – Pertti toimii MelaXPerttinä ja kalibroi jo yhteensovitettuja mallifunktiopaketteja osaksi MELA2.0-järjestelmää. Eräänä päivänä Pertti päättää kalibroida erään funktiokirjaston mallipaketin, jonka kalibroinnista on usean kertaan ollut puhetta kollegan kanssa.

Miten kalibrointi ja funktion hinnan määrittely käytännössä tapahtuu?

Pertti hyödyntää hakukonetta muodostaessaan laskentaketjuja, joita hän tarkastelee työkalun X avulla, että laskentaketju toimii…?

Lopputuloksena syntyy uusi kalibroitu mallipaketti, joka julkaistaan avoimena lähdekoodina ja lisätään funktiokirjastoon.

## Käyttötapaukset

**Käyttötapaus:** Simuloinnin suorittaminen GUI käyttöliittymällä.

**Käyttäjä:** Loppukäyttäjä

**Tavoite:** Simuloida tulosraportti annetulle metsikködatalle.

**Vuorovaikutteiset komponentit:** Käyttöliittymä

**Alustavat toimenpiteet:** MELA2.0-järjestelmän asiakassovellus on asennettu tietokoneelle.

**Toimintavaiheet:**

1. Käyttäjä käynnistää asiakassovelluksen.
2. Käyttäjä valitsee käyttöliittymästä ’Uusi skenaariolaskenta’.
3. Käyttäjä valitsee käyttöliittymästä ’Syötä data’, joka avaa tietokoneen hakemiston.
   1. Käyttäjä valitsee halutun datan skenaariolaskentaa varten.
4. Käyttäjä valitsee ’Tuloskuvauksen muodostaminen’.
   1. Käyttäjä muodostaa tuloskuvauksen.
5. Käyttäjä valitsee ’Syötä käytettävät simulointiskriptit’
   1. ?
6. Käyttäjä tarkastaa yhteenvedon ja käynnistää simuloinnin.
7. Käyttäjä odottaa skenaarioraportin muodostumista.
   1. Käyttäjä tulkitsee skenaarioraportin.
8. Tarvittaessa käyttäjä toistaa vaiheet 4-6, kunnes lopputulos on riittävä.

**Vaihtoehtoiset toimintavaiheet:**

2a. Käyttäjä jatkaa siitä mihin on aikaisemmin jäänyt1.

3a. Käyttäjä vetää datan tietokoneen hakemistosta käyttöliittymän ’tiputus’ -ikkunaan2.

**Huomioitavaa:**

1 Asiakassovellus muistaa viimeisimmän session tilanteen.

2 Tyyppillinen drag’n’drop -alue.

**Käyttötapaus:** Mallipaketin sertifiointi osaksi funktiokirjastoa.

**Käyttäjä:** Kirjastonhoitaja

**Tavoite:** Mallipaketin yhteensovittaminen osaksi MELA2.0-järjestelmää.

**Vuorovaikutteiset komponentit:** Funktiokirjasto, Hakukone

**Alustavat toimenpiteet:** **-**

**Toimintavaiheet:**

1. Mallipaketin vastaanottaminen mallintajalta.
2. Mallipaketin kuittaaminen vastaanotetuksi.
3. Mallipakettiin tutustuminen.
   1. Funktioiden ja metadatan läpikäynti.
4. MELA2.0-yhteensopivuuden mahdollistaminen.
   1. Vaadittujen metatagien lisääminen.
   2. Funktioiden arvoalueiden määrittäminen.
   3. Testausdataan perustuva validointi.
   4. Hakukoneperusteinen testaus.
   5. Vaiheiden 4.1 – 4.4 iterointi kunnes lopputulos vastaa haluttua.
5. Sertifiointi
   1. Sertifiointi metadatan lisääminen pakettiin.
6. Avoimen lähdekoodin julkaisu.
   1. Sertifioidun paketin lisääminen Esim. MELA2.0 GitHub sivulle.
   2. Sertifioidun paketin vieminen CRAN-palveluun.
7. Viestittäminen
   1. Massa sähköpostia sähköpostilistalaisille.
   2. Sähköpostia mallintajalle.

**Vaihtoehtoiset toimintavaiheet: -**

**Huomioitavaa: -**

**Käyttötapaus:** Sertifioitujen mallipakettien kalibroiminen

**Käyttäjä:** MelaXpertti

**Tavoite:** Mallipaketin yhteensovittaminen MELA2.0-järjestelmään

**Vuorovaikutteiset komponentit:** Funktiokirjasto, Hakukone

**Alustavat toimenpiteet:** -

**Toimintavaiheet:**

1. Mallipaketin valinta.
2. Mallipaketin aukaiseminen.
3. Funktion valitseminen.
4. Funktion siirtäminen kalibroitavaksi.
5. Uusien kertoimien kalibrointi.
6. Kalibroitujen kertoimien testaaminen.
7. Kalibroitujen kertoimien hyväksyminen.
8. Kalibroitujen funktioiden siirtäminen uuteen pakettiin.
9. Paketin sulkeminen ja julkaiseminen

**Vaihtoehtoiset toimintavaiheet: -**

**Huomioitavaa: -**

**Käyttötapaus:** Hakukone-komponentin koneoppimisen parantaminen.

**Käyttäjä:** MelaXpertti tai Kirjastonhoitaja

**Tavoite:** Parantaa hakukoneen ymmärrystä laskentaketjujen muodostamisesta.

**Vuorovaikutteiset komponentit:** Hakukone, Funktiokirjasto

**Alustavat toimenpiteet:** Mallifunktioissa on jo alustava hinta-arvo metatieto.

Toimintavaiheet:

1. Testilaskentaketjun1 muodostaminen hakukoneella.
2. Testilaskentaketjun ”aukaiseminen”.
3. Laskentaketjun tulkinta.
4. Hinnan muuttaminen.
5. Uuden laskentaketjun muodostaminen.

**Vaihtoehtoiset toimintavaiheet: -**

**Huomioitavaa:**

1 Vaatii @mustbeincluded -tagin sisällyttämistä funktioihin.

**Käyttötapaus:** Simulointiskriptin valinta ennen simulointia.

**Käyttäjä:** Loppukäyttäjä

**Tavoite:** Valita simulointitarkoitukseen sopiva ohjausskripti.

**Vuorovaikutteiset komponentit:** Käyttöliittymä, MELA2.0-verkkosivut

**Alustavat toimenpiteet:** Käytettävä metsikködata on valittu sekä tuloskuvaus on muodostettu käyttöliittymässä.

**Toimintavaiheet:**

1. Käyttäjä suuntaa MELA2.0-verkkosivulle etsimään erilaisia simulointiskriptejä
   1. Etsii vaihtoehtoisia skriptejä
   2. Lukee skriptien kuvauksia
2. Vaiheet 1.1 – 1.2 toistetaan, kunnes sopiva skripti löytyy.
3. Käyttäjä syöttää käyttöliittymän kautta skriptin tunnistetiedon1.

**Vaihtoehtoiset toimintavaiheet:**

3a. Käyttäjä muodostaa oman skriptin IDE:llä ja syöttää sen käyttöliittymään.

**Huomioitavaa:**

1 MELA2.0 tarvitsee vain tiedon siitä mitä skriptiä käytetään, ei itse skriptiä. Valittu skripti haetaan ennen varsinaista simulointia Valtorin-palvelimelta.

**Käyttötapaus:** Tuloskuvauksen muodostaminen asiakassovelluksessa.

**Käyttäjä:** Loppukäyttäjä

**Tavoite:** Simuloinnin lopputuloksen ja halutun raportin kuvaaminen.

**Vuorovaikutteiset komponentit:** Käyttöliittymä

**Alustavat toimenpiteet:** Käytettävä metsikködata on valittu käyttöliittymässä.

**Toimintavaiheet:**

?

**Vaihtoehtoiset toimintavaiheet:**

**Huomioitavaa:**