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 tutkijat Reetta Lempinen sekä 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 mm.:

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

MELA2.0-metsätietojärjestelmä on tarkoitus syrjäyttää vanha MELA-tietojärjestelmä ja tuoda läpinäkyvyyttä sekä avoimuutta metsäresurssien hyödyntämiseen, niin yksityisille metsänomistajille kuin poliittisille päätöksentekijöille. Lisäksi kuka tahansa voi osallistua järjestelmän kehittämiseen, ylläpitoon ja sisällön tuottamiseen 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 valtion tieto- ja viestintäkeskus Valtorin palvelimilla. Myöhemmässä vaiheessa asiakassovelluksesta on tarkoitus kehittää myös yksinkertaistettu selainpohjainen versio.

Asiakassovellus ei itsessään sisällä laskentaa toteuttavia komponentteja vaan tarjoaa mahdollisuuden välittää dataa varsinaiselle tietojärjestelmälle tarkemmin määritellyn sovellusrajapinnan kautta. Varsinainen Mela2.0-jäjestelmä koostu yhteensovitetuista komponenteista, jotka tarjoavat sovellusrajapintoja hyödyntäen saumattoman laskennan metsikködatalle. Järjestelmän kriittisimpiä komponenttikokonaisuuksia ovat mm. Hakukone, Simulaattori ja Optimoija (komponenteista tarkemmin luvussa 3). Komponenttien toiminnalliset vaatimukset sekä arkkitehtuuri on kuvattu erillisillä dokumenteilla.

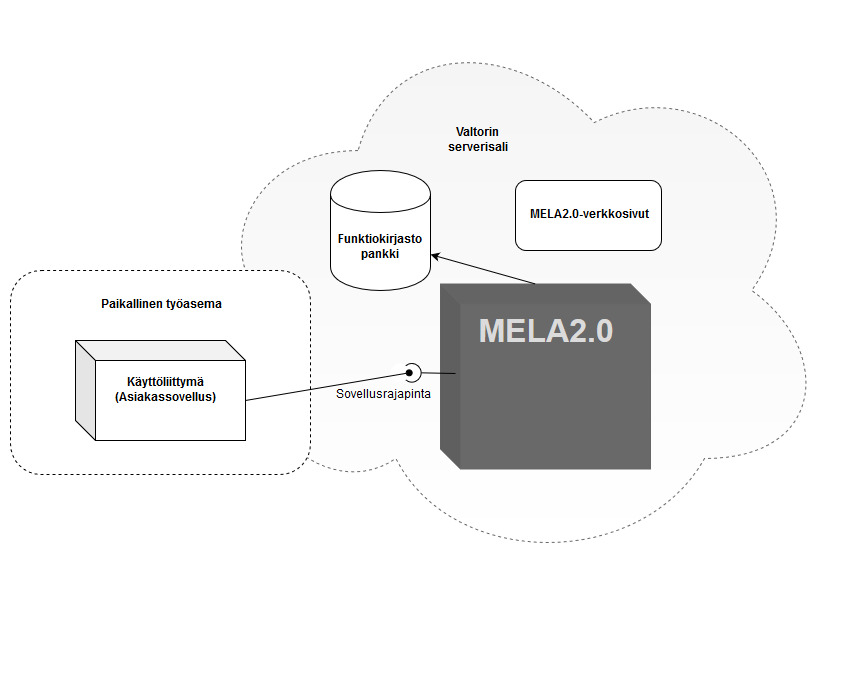
Laskennassa käytettävä metsätieto on sähköisessä muodossa olevaa informaatiota, jonka rakenne voi vaihdella rasterimuotoisesta hiladatasta perinteiseen maastosta kerättyihin havaintoihin. Vaikka käyttäjä voikin määritellä laskennassa käytettävän datan asiakassovelluksessa, on järjestelmän suunnittelussa tunnistettu vaatimus myös integroida ulkoisia datarajapintoja osaksi järjestelmää. Näitä standardoitujen data-aineistoja ovat ainakin valtakunnan metsien inventoinnin metsävara-aineisto (VMI/Luke), Suomen metsäkeskuksen metsävara-aineisto, ja digitaaliset metsänhoitosuositukset (Tapio). Järjestelmän ominaisuuksiin kuuluu annettujen tietojen esiprosessoimaan ja soveltaminen, siten että laskennassa saavutettaisiin optimaalinen tulos. Lisäksi järjestelmä vaatii ulkoiset rajapinnat myös olemassa oleviin mallikirjastoihin kuten Motti, Simo ja Capsis.

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 ovat osa jokaista yksittäistä simulointitapausta.

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

## Järjestelmän yleiskuvaus

Järjestelmän käyttöliittymänä toimii itsenäinen (stand-alone) asiakassovellus. Varsinainen MELA2.0-järjestelmä on sijoitettu Valtorin palvelimelle, joten simulointitoimeksiantojen välittäminen tapahtuu asiakassovelluksella. Valtorin palvelimille sijaitsee myös 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. Käyttöliittymän vaatimusmäärittely löytyy täältä.

**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. Funktiokirjaston vaatimusmäärittely löytyy täältä.

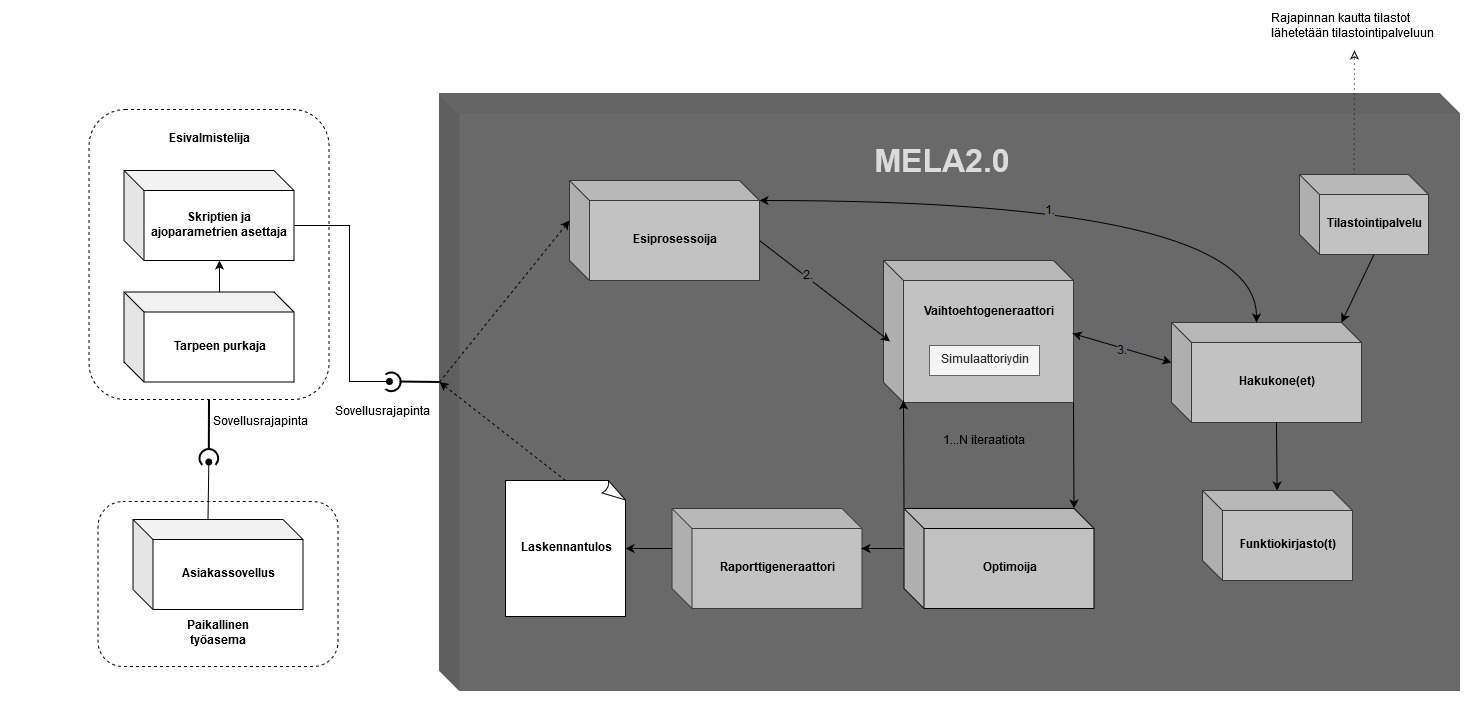
**Esiprosessoija** – Esiprosessoinnin tarkoitus on saattaa syötedata sellaiseen muotoon, että simulointi on mahdollista. Toisin sanoen, jos annetusta datasta puuttuu olennaisia muuttujia, esiprosessointi pyrkii täydentämään puuttuvat muuttujat dataan. Täydennysprosessissa hyödynnetään hakukonetta, joka annettua syötedataa tulkitsemalla tunnistaa ne funktiokirjaston mallifunktiot, joilla esiprosessointi on mahdollista toteuttaa. Esiprosessoijan vaatimusmäärittely löytyy täältä.

**Hakukone** – Hakukone vastaa funktiokirjaston mallipaketeissa olevien funktioiden tulkitsemisesta. Hakukoneen päätarkoitus on etsiä mallifunktioita funktiokirjastosta ja muodostaa laskentaketjuja, joita suoritetaan simulaattorissa tai esiprosessoinnissa. Laskentaketjun muodostus tapahtuu tulkitsemalla mallifunktioiden metakuvauksia. Kuvausten perusteella hakukone sijoittaa funktiot haluttuun kutsujärjestykseen perustuen syötetietona saadulle datalle ja tuloskuvaukselle. Hakukoneen vaatimusmäärittely löytyy täältä.

**Vaihtoehtogeneraattori** – Vaihtoehtogeneraattori on järjestelmän laskennallinen ydin, jonka tarkoitus on toteuttaa varsinainen metsädatalle tapahtuva simulointilaskelma. Vaihtoehtogeneraattori muodostuu vaihtoehtojen generoinnista ja simulointiytimestä. Simulointiydin on geneerinen toteutus, joka pystyy suorittamaan laskennan standardoidulle datan ja hakukoneen muodostavan laskentaketjun avulla. Vaihtoehtoja generoidessaan simulaattori ottaa huomioon ajantasaiset metsänhoitosuositukset ja käyttäjän tuloskuvauksen. Simulaattori tuottaa tuloskuvauksen perusteella käsittely vaihtoehdot, jotka edelleen lähetetään optimointikomponentille. Simulaattorin vaatimusmäärittely löytyy täältä.

**Optimoija** – Simulaattorilta saatu laskennallisesti simuloitu vaihtoehtodata karsitaan siten, että vain todennäköisimmät ja realistisimmat vaihtoehdot säilytetään. Tähän optimoija hyödyntää laskennallista älykkyyttä. Koska simulointioptimointi prosessi on iteratiivinen, on mahdollista, että optimoija käskee simulaattorin toteuttaa uusi simulointi karsitulle vaihtoehtodatalle. Tämä iteratiivinen simulointioptimointi prosessi jatkuu, kunnes optimoija ei pysty enää karsimaan simulaattorin tuottamia vaihtoehtoja. Optimoijan vaatimusmäärittely löytyy täältä.

**Raporttigeneraattori** – Tuloskuvauksen perusteella määräytyy simulointioptimointilaskennan päätteeksi tulostettavat raportit ja tiedostot. Tuloskuvaksen on määritellyt loppukäyttäjä. Raporttigeneraattorin vaatimusmäärittely löytyy täältä.



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

# Järjestelmän käytönkuvaus

## Käyttäjät

Järjestelmän käyttäjiksi on tunnistettu sellaiset sidosryhmänjäsenet, jotka ovat suorassa vuorovaikutuksessa asiakassovelluksen tai jonkun järjestelmän edellisessä luvussa kuvatun komponentin kanssa. Käyttäjäryhmien vuorovaikutus kohdistuu toisistaan poikkeaville komponenteille. Järjestelmän käyttäjät on eritelty neljään pääryhmään: loppukäyttäjät, mallintajat, kirjastonhoitajat ja MelaXpertit. Yksittäinen käyttäjä voi olla edustettuna useammassa ryhmässä. Tietyt käyttäjäryhmät sisältävät alaryhmiä.

Tietojärjestelmän käyttäjäryhmien erittely:

**Loppukäyttäjä** – Järjestelmän loppukäyttäjän päätarkoituksena on simuloida haluamalleen metsikködatalle vaihtoehtoraportteja. Loppukäyttäjän järjestelmän käyttö tapahtuu asiakassovelluksen kautta. Loppukäyttäjät jakautuvat alaryhmiin, joita ovat mm. metsänomistajat, yritykset, ministeriöt, tutkijat ja testaajat.

**Mallintaja** – Mallintajat ovat järjestelmän sisällöntuottajia. Heidän pääasiallinen tarkoituksensa on omien mallipakettien tuottaminen osaksi funktiokirjastoa. Sisällöntuottaminen voi olla paikallista tai julkista. Julkinen sisällöntuottaminen tapahtuu prosessina, johon osallistuu Mela2.0-järjestelmän työryhmä nk. Mela-tiimi. Mela-tiimin sertifioida mallintajan mallipaketteja osaksi Mela2.0-järjestelmää. Näin ollen mallintaja saadaan osallistumaan funktiokirjaston laajentamiseen valitun avoimen lähdekoodin lisenssin mukaisesti. Paikallinen sisällöntuottaminen on lähinnä tutkija orientoitunutta ja vaatii syvällistä ymmärrystä järjestelmän toiminnasta.

**Kirjastonhoitaja** – Vastaa mallipakettien sertifiointiprosessista ja pakettien julkaisemisesta avoimen lähdekoodin lisenssien mukaisesti. Kirjastonhoitaja on osa Mela-tiimiä, joka vastaanottaa mallipaketteja mallintajalta. Tiimin jäsenenä hänen tehtävänsä on validoi, testaa ja lisää tarvittavat metakuvaukset paketin mallifunktioihin. Sertifioitu mallipaketti liitetään osaksi funktiokirjastokomponenttia, jolloin mallipaketti myös julkaistaan avoimena lähdekoodina [GitHub](https://github.com) versionhallintapalvelussa.

**MelaXpertti** – Kuten kirjastonhoitaja, myös MelaXpertti on osa Mela-tiimiä ja vastaa omalta osaltaan mallipakettien sertifioinnista. Xpertti vastaa jo sertifioitujen funktioiden malliparametrien sekä virhekertoimen kalibroinnista. Lisäksi Xpertin tehtäviin kuuluu laatia simulointiskriptejä järjestelmän loppukäyttöä varten.

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

## Sertifiointiprosessi

Mallipakettien sertifiointiprosessin on tarkoitus yhteensovittaa tutkijoiden laskennallisia malleja hyödynnettäväksi Mela2.0-järjestelmässä. Sertifiointiprosessi on osa järjestelmän sisällöntuottamista yhteisöllisen kehittämisen, ts. avoimen lähdekoodin, käytänteiden mukaisesti. Sertifiointiprosessiin osallistuvat järjestelmän käyttäjäryhmistä mallintaja ja MelaXpertistä sekä kirjastonhoitajasta muodostuva Mela-tiimi.

Sertifiointi prosessi saa alkuunsa tutkijan/mallintajan tarpeesta simuloida käsittelyvaihtoehtoja, joissa hän haluaa hyödyntää omia tieteellisiämallejaan. Mela-tiimi vastaanottaa mallintajan mallifunktiopaketin ja toteuttaa sille toimenpiteitä, jotta mallifunktiopaketti voi saada Mela2.0-sertifikaatin. Näitä toimenpiteitä ovat mm. puuttuvien metakuvausten lisääminen, mallien verifiointi ja validointi, funktioiden alustavien hintojen ja rajoitusten määrittely sekä mallin parametrien kalibrointi.

Sertifiointiprosessi on iteratiivinen ja tapahtuu yhteistyössä mallintajan ja Mela-tiimin välillä. Prosessin lopuksi Mela-tiimi julkaisee sertifioidun mallipaketin avoimena lähdekoodina sekä varastoi mallipaketin osaksi funktiokirjasto komponenttia.

Koska mallintajan tehtävä vaatii syvällisempää ymmärrystä järjestelmän toiminnasta, pitää käyttäjäryhmälle tarjota opiskelumateriaalia sekä koulutustilaisuuksia järjestelmän käytöstä ja ajantasaista informaatiota järjestelmän uusista ominaisuuksista.

## Käyttäjätarinat

**Mallipakettien julkaiseminen** –Minna on tutkija, joka on kehittänyt puutason kasvumalleja. Hänellä on kuviotason metsikködataa, jolle hän haluaa simuloida kasvua ja hyödyntää kehittämiään malleja. Minna on kuullut mahdollisuudesta hyödyntää omia malleja simuloinnissa avoimen lähdekoodin Mela2.0-metsikkösimulaattorilla. Hän suuntaa Mela2.0-verkkosivuille tutustumaan Mela2.0:n mallipakettien sertifiointiprosessiin. Tutustuttuaan dokumentaation ohjeistukseen Minna toteuttaa funktiopakettinsa R-kielellä, koska se on MELA2.0:n ensisijainen ohjelmointikieli mallifunktioille. Minna toteuttaa mallipakettinsa ohjeistuksen mukaisesti R-studiolla. Lopuksi hän kuvaa mallifunktiot dokumentaation mukaisella metatiedolla.

Dokumentaation perusteella, Minna toimittaa mallifunktiopakettinsa Mela-tiimille, joka vastaa uusien mallipakettien sertifioinnista ja yhteensovittamisesta osaksi MELA2.0:n funktiokirjastokomponenttia.

Mela-tiimi vastaanottaa Minnan paketin. Tästä alkaa iteratiivinen prosessi, jossa molemmat osapuolet sekä Minna, että Mela-tiimi tekevät aktiivisesti muutoksia ja lisäyksiä malleihin ja niiden metakuvauksiin. Lopuksi mallipaketti saadaan yhteensovitettua osaksi järjestelmää. Mela-tiimi varastoi paketin osaksi funktiokirjastoa ja julkaisee sen avoimen sovitulla lähdekoodin lisenssillä GitHubissa. Nyt Minna tai toinen tutkija voivat hyödyntää Minnan kirjoittamia malleja Mela2.0-simuloinnissa.

**Yksinkertainen simulointi tapaus** – Lauralla on hallussaan metsikködataa XML-muodossa. Hän haluaa simuloida Mela2.0:lla metsikön kasvuennusteen 20 vuoden päähän ja maksimoida tukkipuun tuotannon. Laura käynnistää Mela2.0:n asiakassovelluksen tietokoneeltaan ja valitsee käyttöliittymän avulla XML-tiedoston, joka sisältää laskennassa käytettävän metsikködatan. Seuraavaksi Laura muodostaa tuloskuvauksen, johon hän kuvaa tukkipuun maksimoinnin kahdenkymmenen vuoden päähän simuloinnin lopputuloksena yhteenveto raportissa.

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 simuloinnin aikana 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 varastoi sen osaksi funktiokirjastoa. Varastoiminen tapahtuu siirtämällä mallipaketin lähdekoodi tietokantaan. Lopuksi hän vielä julkaisee mallipaketin versionhallinta-alustalla. Viimeisenä hän informoi mallintajaa/mallintajia siitä, että julkaisuprosessi on nyt valmis ja mallipaketti on saanut MELA2.0-sertifikaatin.

**Funktiokirjaston mallifunktioiden kalibroiminen** – Pertti toimii Mela-tiimissä Mela-asiantuntijana. Hän on vastaanottanut yhdessä kirjastonhoitajan kanssa mallintajan mallipaketin. Pertti tutustuu malleihin ja suunnittelee validointi-, verifiointi- ja kalibrointiskenaariot erilaisille metsikködatoille. Hän määrittelee hinnoittelun mallifunktioille ja kalibroi jokaisen mallifunktiopaketin mallifunktion erikseen. Lopuksi hän tallentaa kalibroinnin tuloksen mallifunktiopakettiin omaksi tiedostokseen. Kalibroidut parametrit ja kyseisen mallin hän yhdistää metakuvauksella.

## 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:**