

SAVONIA

ammattikorkeakoulu

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

PROMS MAST-2 SIMULAATIOYMPÄ- RISTÖ

TEKIJÄ Mikko Rautava

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä(t) Mikko Rautava	
Työn nimi ProMS MAST-2 simulaatioympäristö	
Päiväys 16.4.2024	Sivumäärä/Liitteet 25
Toimeksiantaja/Yhteistyökumppani(t) Arbonaut Oy	
<p>Tiivistelmä (Huom. kirjoita teksti alla näkyvään harmaaseen kenttään; huomioi tämä myös kopioitaessa)</p> <p>Tässä opinnäytetyössä perehdyttiin ProMS:n projektiympäristöön sekä MAST-2 hankkeen simulaatioympäristöön. Työn taustalla oli luoda Suomen pelastuslaitosten käytössä olevaan skenaario-simulaatiotyökaluun laajennos, jota voitaisiin hyödyntää maastopalojen suunnittelussa ja torjunnassa. MAST-hankkeiden tavoitteena on ollut kehittää simulaatioympäristöä, kehittämällä projektiympäristö, jossa on kattavat työkalut simulaa-tiotulosten tarkasteluun.</p> <p>Työssä hyödynnettiin suunnittelumenetelmiä ja ohjelmointitaitoja laajennoksen kehittämisessä. Alkuvaiheessa suunniteltiin simulaatiotyökalun kehityssuunnitelma, joka toteutui 2023. Suunnitelman jälkeen suoritettiin perusteellinen tutkimus ProMS:n järjestelmän projektiympäristöstä ja hankkeen MAST-2 kehitysympäristöstä, tämä osuus toteutettiin 2024.</p> <p>Opinnäytetyöllä kasvatettiin ymmärrystä simulaatioympäristössä sekä ProMS:n kehitysympäristössä toimimisesta. Vaikka kehitystyö jäi keskeneräiseksi, se tarjosi arvokasta tietoa ja pohjaa tuleville kehitystoimille ProMS:n järjestelmäympäristössä.</p>	
Avainsanat simulaatioympäristö, projekti, palotorjunta, maastopalo	

Field of Study Technology, Communication and Transport	
Degree Programme Degree Programme in Information Technology	
Author(s) Mikko Rautava	
Title of Thesis ProMS MAST-2 Simulation Environment	
Date 14 May 2024	Pages/Appendices 25
Client Organisation /Partners Arbonaut Oy	
<p>Abstract</p> <p>The aim of the thesis was to study the ProMS project environment and the simulation environment of the MAST-2 project. The background for this thesis was to create an extension for the Finnish Fire Department simulation scenario tool that was already in use in the simulation environment. The extension could further help with the planning of extinguishing the wildfires. The goal of the MAST projects has been to develop a comprehensive simulation environment for analyzing simulation results.</p> <p>Design methods and programming skills were utilized in the development of the tool extension. In the initial phase, a plan was drawn up for the extension of the tool in 2023. After the plan was finalized, a fundamental study of the ProMS development environment and the project environment MAST-2 was carried out in 2024.</p> <p>The thesis increased understanding in working with the simulation environment and working with the ProMS development environment. The development work was not completed but thesis still provided important knowledge and skills for future ProMS project development.</p>	
<p>Keywords</p> <p>simulation environment, project, fire prevention, wildfire</p>	

SISÄLTÖ

1	JOHDANTO	7
1.1	Työn tausta ja tavoite	7
1.2	Arbonaut Oy	7
2	REACT	8
3	REDUX	10
4	PROMS	11
5	SIMULAATION KÄYTTÖTARKOITUS	12
6	SIMULAATIOYMPÄRISTÖ	13
6.1	Prometheus	13
6.2	Simulaatioympäristön karttatasot	13
6.3	Simulaatiotyökalut.....	14
6.4	Simulaatiopyynnöt	15
6.4.1	FMI-ARBO rajanpinta	15
6.4.2	Simulaatiopyynnön rakenne.....	15
6.5	Simulaatitulokset.....	17
7	TYÖKALUN TOTEUTUS.....	20
7.1	Työkalun suunnittelu	20
7.2	Työkalun toteutus	21
7.3	Työkalun testaus.....	22
8	POHDINTA.....	23
8.1	Työn suunnittelu	23
8.2	Simulaatioympäristö.....	24
	LÄHTEET	25

KUVALUETTELO

Kuva 1 - Esimerkki React applikaatiosta	8
Kuva 2 - React komponentin rakenne	9
Kuva 3 - React applikaation rakenne	9
Kuva 4 - Redux toiminta	10
Kuva 5 - ProMS projektimäärittäminen	11
Kuva 6 - Polttoainetyyppikartta	13

Kuva 7 - Syttymisherkkyys.....	14
Kuva 8 - Korkean tason prosessi, simulaation käynnistämiseen	15
Kuva 9 - Simulaatiopyynnön rakenne.....	16
Kuva 10 - Simulaatiopyyntöjen rakenteita	16
Kuva 11 - Simulaatio tulos: SUCCESS	17
Kuva 12 - Simulaation ensimmäinen tunti	17
Kuva 13 - Prometheus: palosimulaatio.....	18
Kuva 14 - Prometheus: palon intensiteetti	18
Kuva 15 - Prometheus: Liekin pituus karttataso.....	19
Kuva 16 - Figma, teemaluonnos ja skenaarion tilat.....	20
Kuva 17 - Figma valikkoluonnos	20
Kuva 18 - Skenaarion valinta työkalulla.....	21
Kuva 19 - Polttoainemuutoksen valikko.....	22
Kuva 20 - Redux-logger	22

Avainsanoja

FMI = Finnish meteorological institute, eli Ilmatieteenlaitos.

Simulaatioympäristö = Projektiympäristö, josta tehdään simulaatiopyyntöjä ja tarkastellaan simulaation tuloksia.

Komponentti = Osa tai kokonaisuus, joka toteuttaa toiminnallisuutta.

WFS = Web Feature Service eli rajapintamäärittely. Tässä työssä se tarkoittaa karttatasoa, jonka vektoritasot ovat tarkasteltavissa ja ne ottavat vastaan parametrejä.

Skenaario = Tässä yhteydessä sillä tarkoitetaan tapahtumaketjua, joka on riippuvainen annetuista lähtötiedoista (simulaation lähtötiedot vaikuttavat simulaatio tulokseen).

1 JOHDANTO

1.1 Työn tausta ja tavoite

Työn tarkoituksena on tutustua MAST-2 hankkeen simulaatioympäristöön sekä tutustua ProMS:n kehitysympäristöön. MAST-2 on hanke, jonka tarkoituksena on tuottaa maastopalojen leviämismalleja pelastustoimen tilannekuvan muodostamisen tueksi. Tämä on jatkohanke hankkeelle MAST. Työssä myös suunnitellaan lisätoiminnallisuus hankkeen simulaatiotyökaluun, jolla voitaisiin tehostaa simulaatioiden käytettävyyttä antamalla mahdollisuus luoda skenaarion valinnat yhtenäisen työkalun kautta. Simulaation tulosten tarkastelu on näin helpompaa ja selkeämpää käyttäjille. Työssä perehdytään tekniikoihin, joita hyödynnetään ProMS:n tietorakenteessa ja arkkitehtuurissa. Opinnäytetyö on tehty toimeksiantona Arbonautille Oy:lle.

Tässä työssä käsitellään simulaatioympäristössä toimivaa projektikokonaisuutta. Työn tavoitteena on opetella toimimaan ProMS:n kehitysympäristössä, ymmärtää simulaatioympäristön toiminnallisuutta sekä suunnitella ja toteuttaa laajennos simulaation skenaariotyökalun toiminnallisuuteen. Tällä laajennoksella parannetaan skenaarioiden rakentamisesta käyttäjäystävällisempää.

1.2 Arbonaut Oy

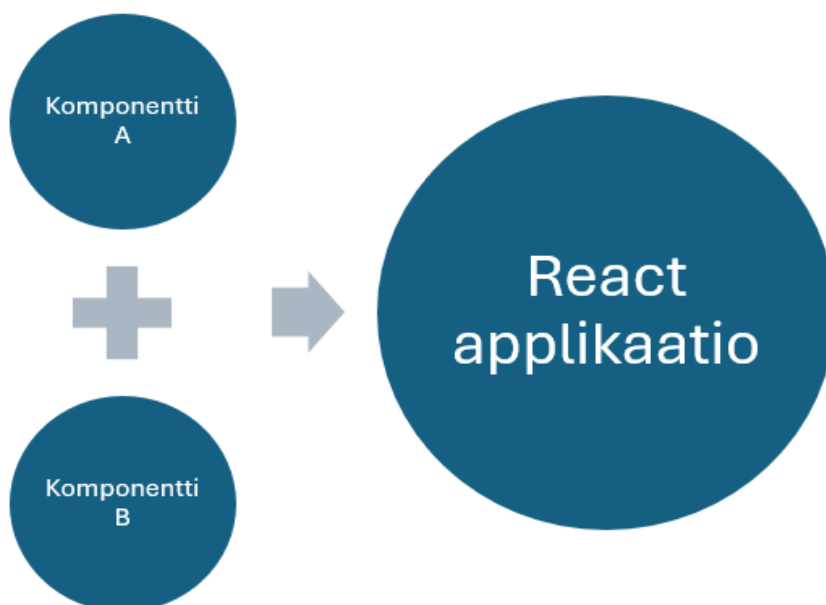
Arbonaut Oy on tietotekniikan- ja luonnonvara-alan yritys ja se on perustettu vuonna 1994. Arbonaut on omistautunut tarjoamaan yksilöllisiä ratkaisuja asiakkaiden tarpeisiin. Se toteuttaa hankkeita ja projekteja asiakkaille esimerkiksi päätöksenteon tueksi. Tässä työssä käsiteltävä simulaatioympäristö on rakennettu päätöksenteon tueksi ja pohjaksi se hyödyntää monipuolisesti Arbonautin tuottamia paikkatietoaineistoja sekä Prometheus metsäpalosimulaattorin tuottamia ennustemalleja. (Arbonaut 2024.)

2 REACT

React on JavaScript-kirjasto, joka käyttää komponentteja käyttöliittymän rakentamiseen. Komponentti voi sisältää tekstilaatikon, painikkeen, koko lomakkeen tai vaikka joukon muita komponentteja. Esimerkiksi koko sovelluksen käyttöliittymä voi olla yksi komponentti. Tämä mahdollistaa komponenttien uudelleen käytettävyyden, kun ne ovat eristettyjä toiminnallisuuksia ja kokonaisuuksia. (Chiarelli 2018, luku 1.)

Komponenteilla on oma tila, jolla tarkoitetaan ominaisuuksia, jotka ohjaavat komponenttien käyttäytymistä. Tilaan määritetään tieto mitä komponentti käyttää. Tätä tietoa muuttamalla ohjataan komponentin toimintaa sen elinkaaren aikana, jota hyödynnetään toiminnan tai ulkoasun hallinnassa. (Geeks for Geeks 2024.)

Esimerkki React applikaation rakenteesta. Kuvassa (ks. kuva 1) React applikaatio muodostuu komponentti A:sta sekä komponentti B:stä, ja nämä muodostavat yhdessä käyttöliittymän eli React applikaation. Komponenttia A voitaisiin lisätä applikaatioon haluttu määrä koska komponentit ovat kokonaisuuksia, joita voidaan uudelleen käyttää. Esimerkissä voimme kuvitella, että esimerkiksi komponentti A sisältää käyttöliittymän ja B sisältää laskurin.



Kuva 1 - Esimerkki React applikaatiosta

Kuvassa (ks. kuva 2) on yksinkertainen laskurikomponentti, joka ottaa vastaan ominaisuustietoja. Ominaisuustiedot tässä esimerkissä sisältävät vain tekstiä. Laskuri ottaa ominaisuustiedon nimeltä "arvo", joka annetaan laskurille sen alustusvaiheessa (ks. kuva 3). Tätä tietoa käytetään komponentissa laskurin nimeämiseen (ks. kuva 2). Laskurin tilaa muutetaan aina, kun käyttöliittymässä painetaan laskurin "Lisää"-nappia. Tällä tavoin komponentin tilaa muuttamalla myös käyttöliittymässä laskurin näyttämä tieto muuttuu.

```
import React from 'react';
import { useState } from 'react';

function Komponentti(props) {
  //funktio ottaa parametrikksi props, eli ominaisuustiedot

  //state hallitsee komponentin tilaa
  const [maara, setMaara] = useState(0);

  //komponentti käyttää props.arvo ominaisuustietoa laskurin nimeämiseen
  return (
    <div>
      <p>{props.arvo}: {maara}</p>
      <button onClick={() => setMaara(maara + 1)}>Lisää</button>
    </div>
  );
}

export default Komponentti;
```

Kuva 2 - React komponentin rakenne

```
import './App.css';
import Komponentti from './Komponentti';

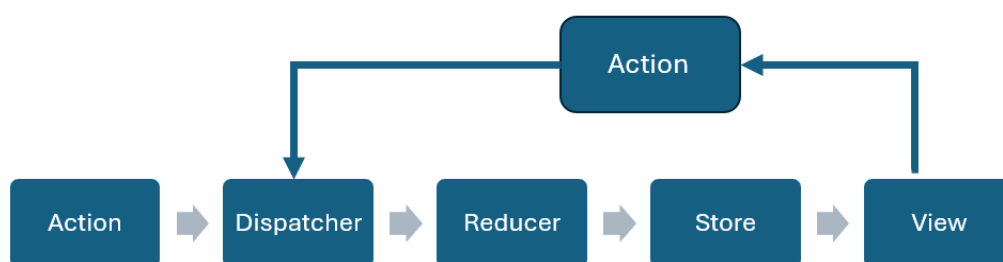
function App() {
  return (
    <div className="App">
      <Komponentti arvo={"laskuri"}/>
      <Komponentti arvo={"toinen laskuri"}/>
    </div>
  );
}

export default App;
```

Kuva 3 - React applikaation rakenne

3 REDUX

Redux on malli ja kirjasto, jonka toiminnallisuus perustuu tilojen muuttamiseen ja hallintaan. Näitä tiloja päivitetään käyttämällä actioneita eli toimintoja. Tämä helpottaa laajojen applikaatioiden hallintaa, sillä kaikki tilapäivitykset tapahtuvat keskitetysti Reduxin lisäämässä store-ominaisuudessa, jossa kaikki tilamuuttujat ovat. Storen tehtävä on vastata applikaation nykytilasta ja ainoa tapa päivittää storen tilamuuttujia on käyttää Redux:n toimintoja. Nämä toiminnot ohjataan dispatcherille eli toimintojen ohjaajalle, jotka taas ohjaavat pyynnöt reducereille eli vähentäjille. Vähentäjät vastaavat oikean tilapäivityksen tekemisestä. Tämän jälkeen storen tila sekä applikaation näkymä päivittyvät (ks. kuva 4). (Redux 2024.)



Kuva 4 - Redux toiminta

4 PROMS

ProMS eli Project Management System on Arbonautin kehittämä projektienhallintasovellus, jonka arkkitehtuuriin kuuluu Redux sekä React. ProMS:n tilanhallinta tapahtuu Reduxin avulla projektikohtaisesti. Jokaisella projektilla on oma Redux store, joka otetaan käyttöön projektia valittaessa. Reactia käytetään käyttöliittymän rakentamisessa ja ydintoiminnot ovat järjestelmän toteuttamia komponentteja. ProMS tukee monia erilaisia projekteja ja jakaa niiden kanssa yhteisiä komponentteja.

ProMS:n projektienhallinta toimii siten, että projektien määrittelyt ovat tehty projektikohtaisesti. Tämä tarkoittaa sitä, että ProMS:sta voidaan valita haluttu projekti, mikä hakee projektikohtaiset määrittelyt (ks. kuva 5). Nämä määrittelyt tulevat voimaan, kun projektitasoa vaihdetaan. Tämä mahdollistaa sen, että kaikilla paikkatietopohjaisilla projekteilla on ydintyökalut helposti käytettävissä, eikä jokaiseen projektiin tarvitse erikseen rakentaa perustoiminnallisuuksia, vaan ne pitää vain määrittellä projektikohtaisesti.



Kuva 5 - ProMS projektimäärittely

5 SIMULAATION KÄYTTÖTARKOITUS

Simulaatiolla pyritään luomaan realistinen ennuste metsäpalon leviämisestä annetun syttymiskohdan perusteella. Simulaatio hyödyntää Prometheus-mallin matemaattisia kaavoja, jotka huomioivat myös ympäristön vaikutukset palon etenemiseen, kuten tuulensuunnan, ilmankosteuden ja paloaineksen määrän.

Projektialustalla voidaan lisätä simulaation luomiselle kriittistä tietoa ympäristöstä, kuten esimerkiksi palokatkot, palosillat sekä avohakkuut, jotka vaikuttavat simulaation tulokseen. On oleellista, että simulaation tulokseen vaikuttavat lähtötiedot ovat paikkansapitäviä, muuten simulaation tulos antaa virheellisen kuvan palon etenemisestä alueella. Simulaation lähtötiedot eivät aina ole ajantasaisia, minkä vuoksi niitä voidaan korjata alustan lisätyökaluilla.

Simulaation tuloksilla pyritään ennakoimaan metsäpalon leviämisen laajuus sekä selvittämään, mil-laisilla toimenpiteillä palon laajuutta pystytään rajoittamaan. Tämä on erittäin tärkeää palontorjun-nan näkökulmasta, sillä näin kriittistä palokalustoa voidaan sijoittaa oikeaan paikkaan, oikeaan ai-kaan. Simulaatioalustalle voidaan myös lisätä esimerkiksi vedenottopaikkoja, jotka auttavat operaa-tiotoiminnan suunnittelussa alueella.

Projektialustalle voidaan kirjata todellinen paloala ja vertailla miten simuloitu paloala eroaa todelli-sesta paloalueesta. Todellisen paloalueen tärkeys korostuu simulaatioympäristössä, koska näin voi-daan verrata tapahtunutta ennusteeseen, joka antaa kriittistä lisäarvoa simulaation oikeellisuudelle. Tätä tietoa hyödyntämällä mallia pystytään jatkokehittämään, jotta tulevaisuudessa palomallien tarkkuus olisi parempi.

6 SIMULAATIOYMPÄRISTÖ

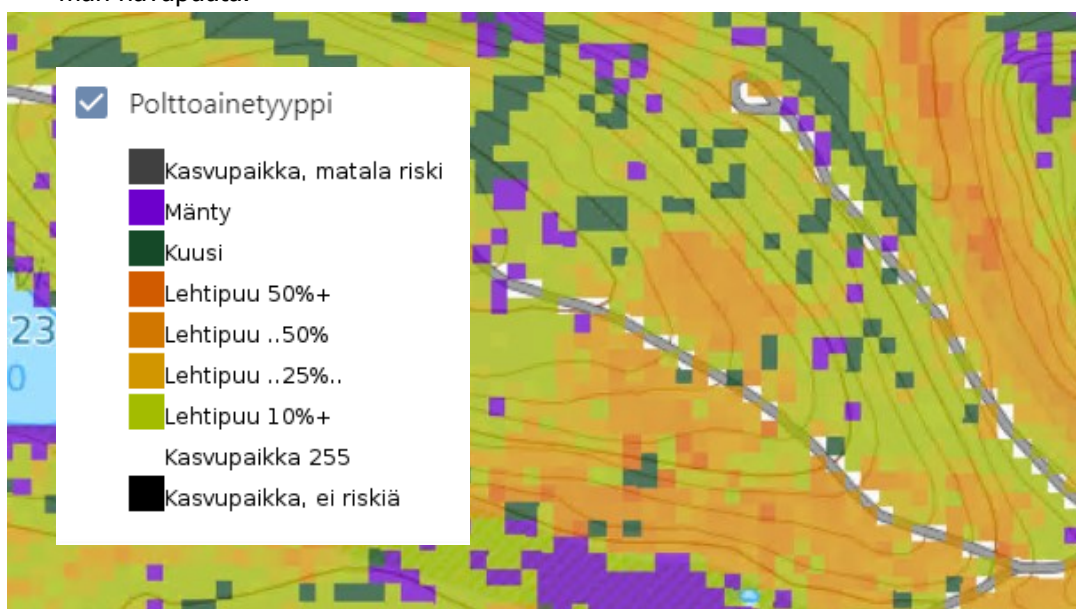
6.1 Prometheus

Prometheus on kanadalainen simulaatiomalli maastopalojen simuloimiseen. Se on deterministinen palonkasvun malli, joka on alun perin kehitetty auttamaan ja ymmärtämään todennäköisiä syyseuraussuhteita paljontorjunnan päätöksenteossa. Malli hyödyntää Huygenin periaatetta maastopalon ennustamiseen, jossa huomioidaan paloaine sekä säätiedot. Malli tuottaa yksityiskohtaisesti palon etenemisprosessin käyttäjän määrittämällä aikaväleillä. (Tymstra 2010, sivu 7.)

PSaaS (Prometheus Software as a Service) on simulaatioympäristön taustalla, joka generoi simulaatiopyynnöille tuloksen. Prometheus hyödyntää erilaisia matemaattisia malleja simulaation luomisessa, sekä erinäisiä ympäristötietoja kuten ilmanlämpötilan, ilmankosteuden, tuulen suunnan, paloaineen määrän.

6.2 Simulaatioympäristön karttatasot

Simulaatioympäristön karttatasot sisältävät projektiympäristön tarjoamat karttatasot, jotka eivät vaadi simulaation tekemistä niiden tarkastelemiseksi. Näiden karttojen tarkoituksena on antaa pohjatietoa ympäristöistä, joihin potentiaalisia simulaatioita voidaan alustaa. Polttoainetyyppi karttatason polttoaineella tarkoitetaan puumassaa ja puulajit ovat lajiteltu polttoaineluokkiin (ks. kuva 6). Kartasta voidaan tarkastella millä alueella on esimerkiksi runsaasti lehtipuuta ja missä taas enemmän havupuuta.



Kuva 6 - Polttoainetyyppikartta

Syttymisherkkä karttataso, joka sisältää syttymisherkyyden, eli kuinka helposti alueella syttyy maastopalo (ks. kuva 7). Tällä karttatasolla voidaan katsoa potentiaalia paloherkkiä alueita.



Kuva 7 - Syttymisherkkyys

Ilmatieteenlaitos tarjoaa Prometheus simulaatioon tarvittavat säähän liittyvät laskentatiedot. MAST-2 projektissa käytettävät ilmatieteenlaitoksen tarjoamat karttatasot ovat mm. lämpötilaennuste, sademääräennuste, ilmankosteusennuste, ukkosennuste, tuulennopeusennuste, tuulensuuntaennuste ja metsäpaloindeksi. Näitä voidaan käyttää simulaation tueksi, jos halutaan tehdä simulaatio esimerkiksi tuulisena päivänä. Tällöin voidaan käyttää tuulennopeusennustetta pohjatietona simulaation alustukselle.

6.3 Simulaatiotyökalut

Simulaatiotyökalujen tarkoitus on helpottaa simulaatituloksen tarkastelua sekä antaa siihen vaikuttavia parametrejä. Parametreillä tarkoitetaan tässä yhteydessä polttoaineluokkien muutoksia, jotka piirretään kartalle vektorimuodossa. Itse simulaatio tulee Prometheusesta, joka prosessoinnin jälkeen palautuu tarkasteltavaksi WFS-karttatasona MAST-2 projektille ProMS:iin.

Pistetiedot ja muut polttoaineluokkiin vaikuttamattomat pistegeometrit sekä vektoriaineistot ovat tarkoitettu lisäämään informaatiota simulaatituloksen tueksi. Esimerkiksi kartalle voidaan lisätä vedenottoaikka, joka ei vaikuta suoraan simulaation tulokseen, mutta helpottaa simulaation hyödyntämistä käytännön ympäristössä.

Simulaatiopyyntö-työkalulla annetaan simulaatiolle alkupiste mihin mahdollinen palosimulaatio luodaan. Simulaation onnistuminen vaatii sen, että alue on tarpeeksi hyvin syttyvää. Sitä ei voi alustaa esimerkiksi veden päälle, koska se ei ole palavaa materiaalia.

Simulaatiopyyntö lisätään kartalle pisteaineistona. Sille lisätään myös aloitusajankohta, joka vaikuttaa simulaatioon siten, että simulaatio hyödyntää säätiedotteita annetusta paikasta ja ajasta. Optimaalinen tilanne simulaation alustamiseen on kuiva ja hyvin paloainesta sisältävä paikka.

Polttoainemuutokset lisätään kartalle polygoniaineistona, jotka huomioidaan simulaatiopyynnössä. Näitä voi lisätä haluamansa määrän ja niiden vaikutus on määritetty polttoaineluokilla, jotka ovat

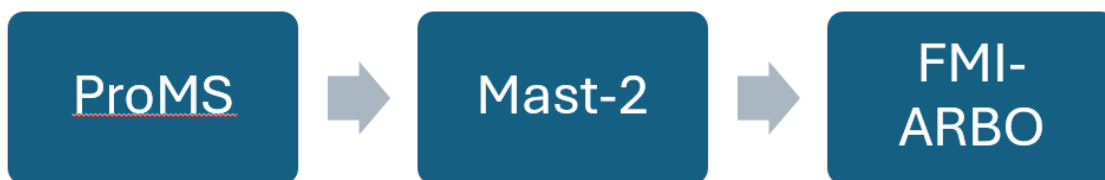
määritelty ProMS:iin. Polttoainemuutoksille voidaan antaa erilaisia luokkia: Aukot, taimikot, palokat-
kot sekä palosillat.

Simulaation tulokseen vaikuttamattomia työkaluja ovat seuraavat: Vesiresurssit, Miehistö-/kalusto, Todellinen paloala. Vesiresurssit-työkalulla lisätään vesiresurssit vektoriaineistona kartalle, nämä ei-
vät vaikuta simulaation lopputulokseen. Vesiresurssien tarkoituksena on merkitä kartalle esimerkiksi vedenotto paikkoja helikopterille. Miehistö-/kalusto- työkalulla lisätään miehistö/kalusto kartalle pis-
teaineistona. Työkalun tarkoituksena on helpottaa operaatiotoimintaa. Tällä työkalulla voidaan lisätä
esimeriksi paloautojen sijainti operaatiosuunnittelua varten. Todellinen paloala-työkalulla lisätään
todellinen paloala kartalle vektorina, ja sillä voidaan merkata kartalle todellinen maastopaloalue.
Työkalun tarkoituksena on verrata simulaation tulosta tapahtuneeseen maastopaloon. Tämä auttaa
hahmottamaan simulaation ja oikean tapahtuneen metsäpalon eroavuuksia.

6.4 Simulaatiopyynnöt

6.4.1 FMI-ARBO rajanpinta

Simulaatiopyyntöjen lähetyk tapahtuu FMI-ARBO rajapinnan kautta MAST-2 projektissa (ks. kuva 8).
FMI-ARBO on Ilmatieteenlaitoksen ja Arbonautin yhteinen rajapinta. Rajapinta on määritelty siten,
että kaikki tarvittavat tiedot koostetaan JSON muotoisesti simulaatiopyyntöön, jonka Ilmatieteenlai-
tos välittää Prometheukselle. Simulaation käynnistäminen palauttaa MAST-2 projektiin simulaation
tilan.



Kuva 8 - Korkean tason prosessi, simulaation käynnistämiseen

Simulaation tilan tehtävänä on kertoa missä vaiheessa simulaatio on. Mahdollisia vaiheita on esimer-
kiksi tekeytymässä, keskeytetty sekä virhe. Kun simulaatio on valmis, voidaan simulaation tulos ot-
taa tarkisteluun WFS-tasolla. Tasolle voi antaa parametriksi ajanjaksoja, joilla voidaan tarkastella
haluttua aikaväliä.

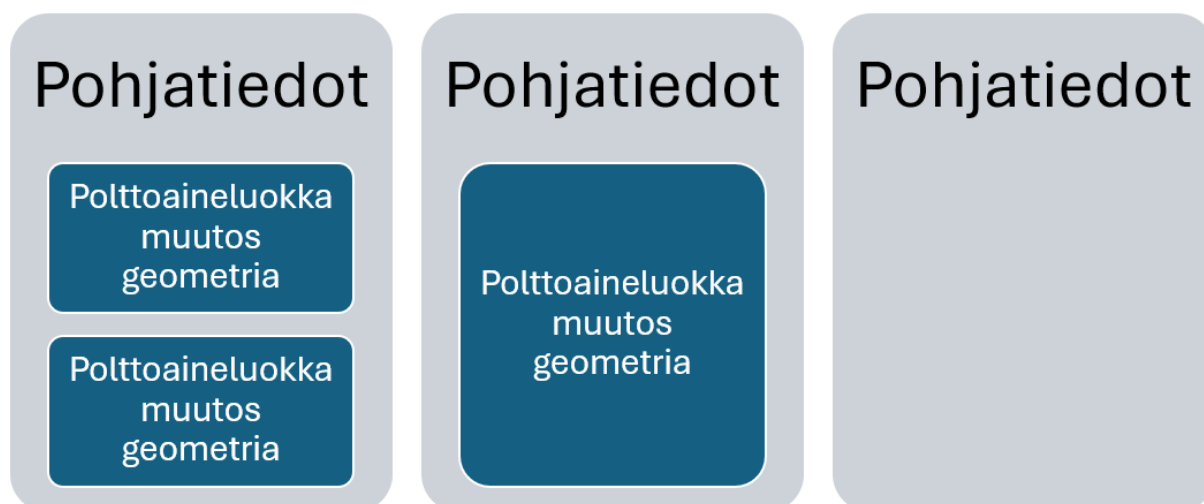
6.4.2 Simulaatiopyynnön rakenne

Simulaatio muodostuu kahdesta osasta: simulaation pohjatiedoista sekä polttoainemuutoksista (ks.
kuva 9). Polttoainemuutokset ovat valinnaista lisätietoa. Niihin lisätään tiedot simulaation omista-
jasta, skenaariosta sekä vektorigeometriasta, joka sisältää polttoaineluokkamutokset. Polttoaine-
muutokset lisätään ennen simulaatiopyyntöä. Simulaation pohjatietoihin kuuluvat: projektio, aloitus-
aika, lopetusaika, sekä pistegeometria. Simulaation pituudeksi valitaan 24 h, jos loppuajankohtaa ei
anneta erikseen. Pistegeometria on simulaatiopyynnön aloituspiste, josta simulaatio halutaan aloit-
taa.



Kuva 9 - Simulaatiopyynnön rakenne

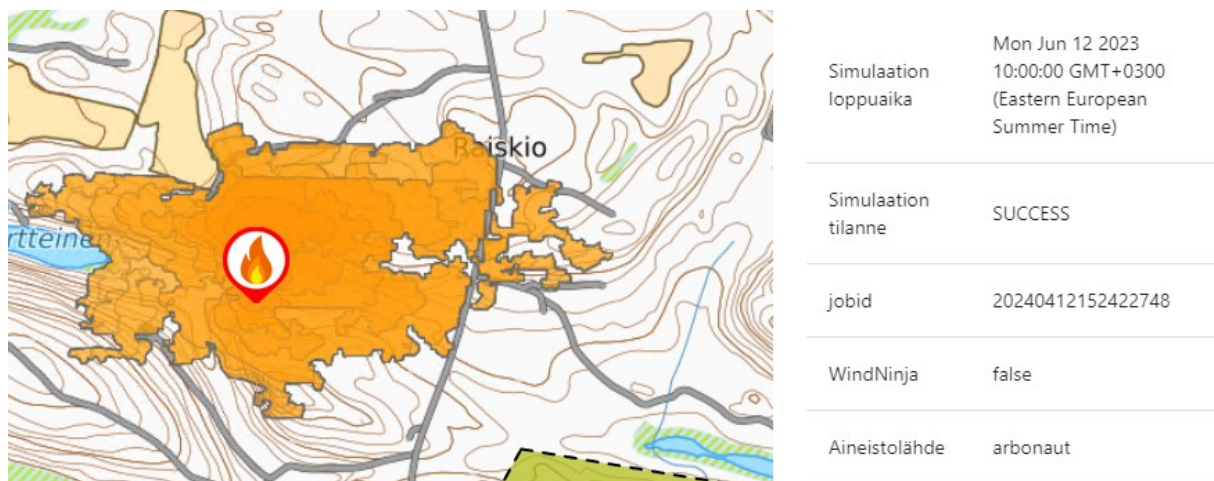
Simulaatiopyynnot toteuttavat myös skenaariotoimintoa, joka tarkoittaa sitä, että yhteen simulaatiopyyntöön pystytään lisäämään useampi polttoainemuutos. Nämä yhdistetään simulaatiopyyntöön ja näitä polttoainemuutoksia kutsutaan skenaarioiksi. Yhden simulaatiopyynnön mukana voi tulla monta skenaariota. Jokaiselle skenaariolle simuloidaan oma tulos, joka palautuu WFS-tasolla, kun simulaatio on valmis. Skenaarioiden luominen tapahtuu lisäämällä polttoainemuutoksia simulaatiopyyntöille. Kuvassa (ks. kuva 10) on kolme esimerkkiä pyyntöjen rakenteesta, jokaisessa pyynnössä on oltava pohjatiedot, mutta polttoainemuutos geometrioita voi olla useampia.



Kuva 10 - Simulaatiopyyntöjen rakenteita

6.5 Simulaatiotulokset

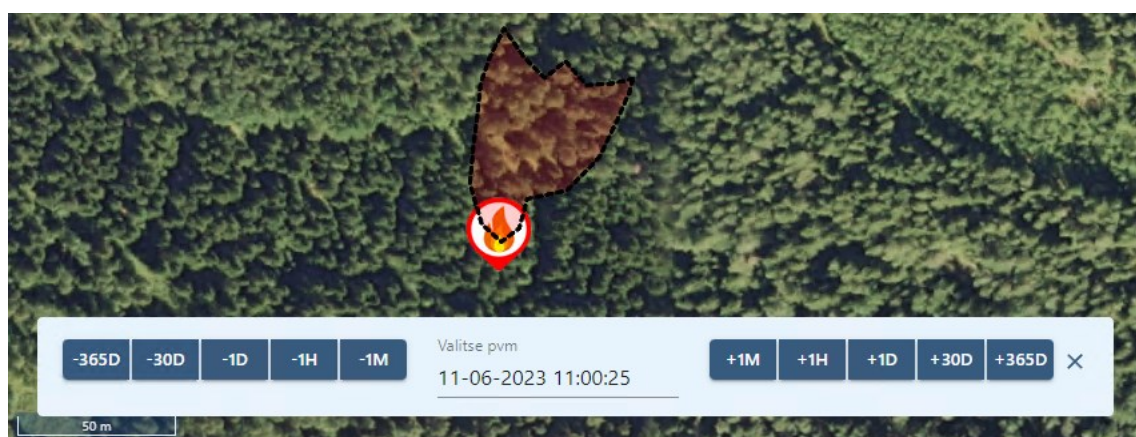
Simulaatiotulokset ovat WFS-tasolla ja niitä pystytään tarkistelemaan ProMS:ssa karttatasona. Kun simulaatio on valmis, simulaation tilaan SUCCESS (ks. kuva 11). Tämä tarkoittaa sitä, että WFS-taso on nyt tarkasteltavissa projektialustalla.



Kuva 11 - Simulaatio tulos: SUCCESS

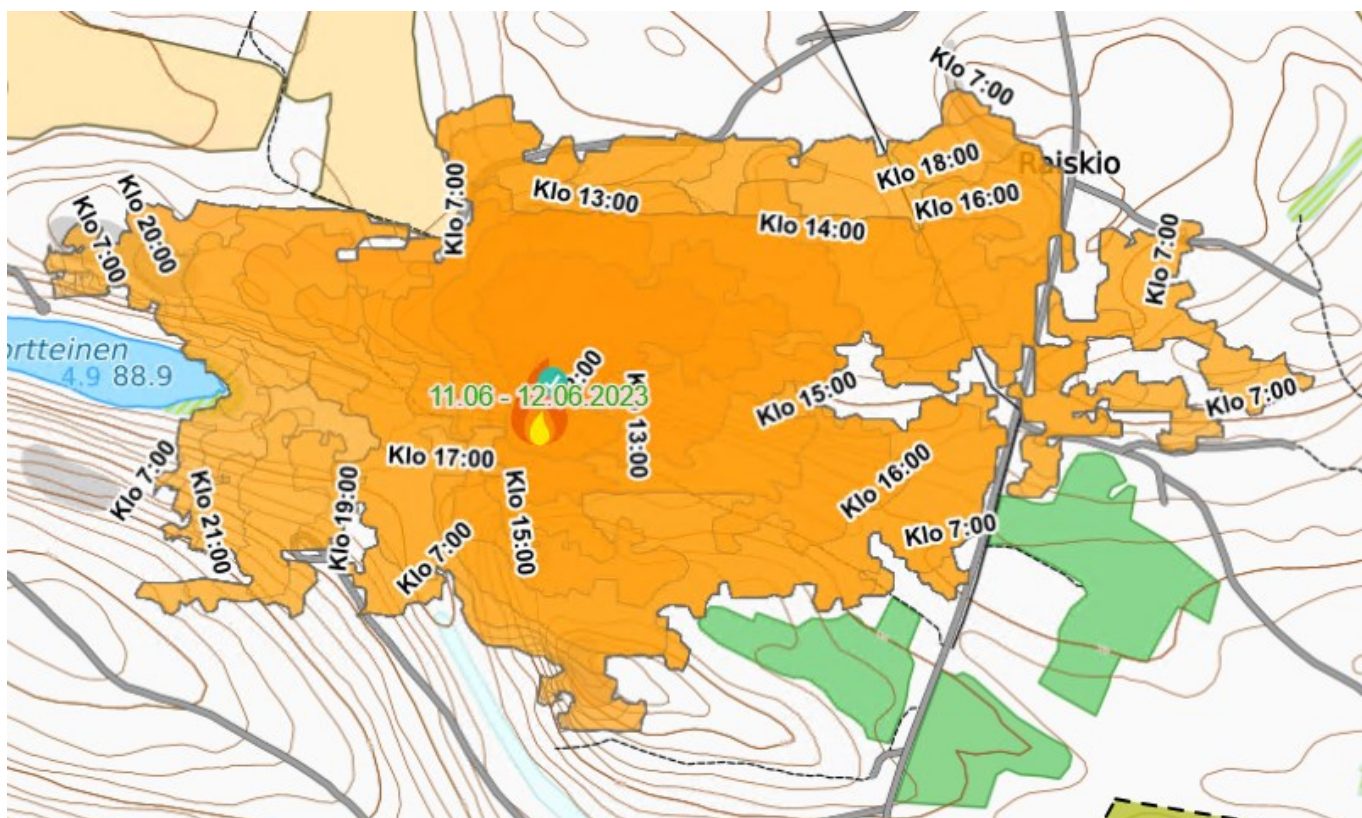
Kuvassa (ks. kuva 12) näkyy simulaation ensimmäisen tunnin palon etenemisalue. Simulaatio on alustettu alkamaan 11.6.2023 kello 10:00 ja sille ei ole annettu lopetusajankohtaa. Kuvassa (ks. kuva 12) näkyy tuntityökalu, jolla voidaan seurata palon etenemistä. Simulaation etenemisintervallit ovat yhden tunnin välein, jolloin simulaatiossa on vakiona 24 eri aikaviipaletta, jotka edustavat jokaisen tunnin palon etenemisaluetta.

Tarkasteluvaiheessa simulaatiolla on karttatasoina palosimulaatio, palon intensiteetti sekä liekinpituus. Nämä tasot tulevat simulaation WFS-tasosta ja ovat tarkasteltavissa, kun simulaatio on onnistuneesti tehty.



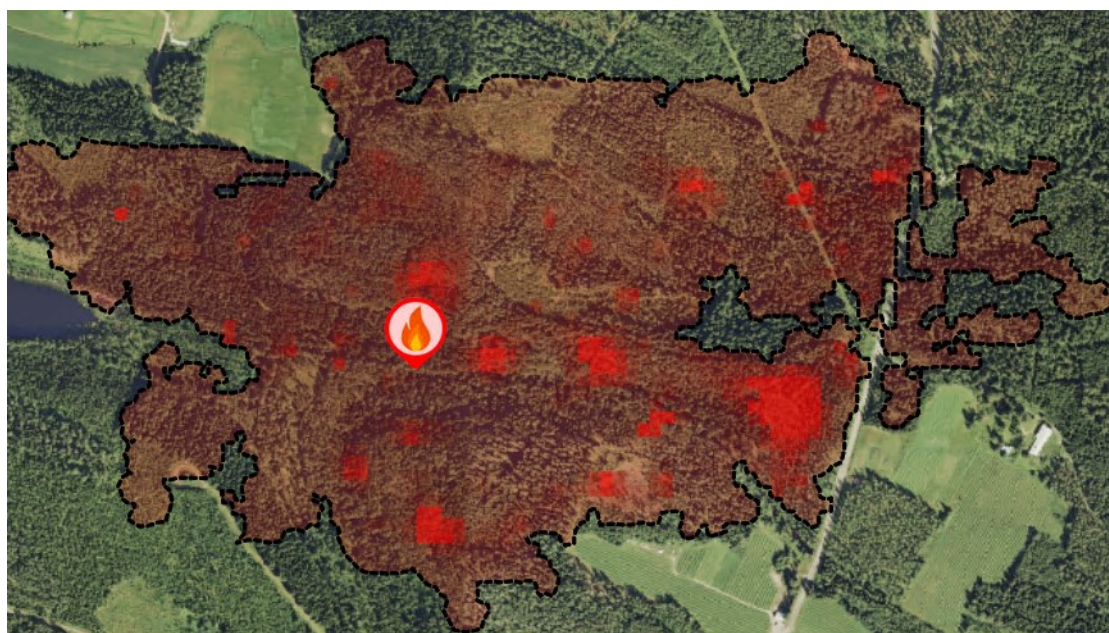
Kuva 12 - Simulaation ensimmäinen tunti

Koko simulaation paloalueen tuloksena on palosimulaatiokartta. Tämä karttataso kertoo koko simulaatio alueen ja aikaleimoittain palon etenemisalueet. Kartasta saa hyvän yleiskuvan paloalueesta ja sen etenemisajankohdista (ks. kuva 13).



Kuva 13 - Prometheus: palosimulaatio

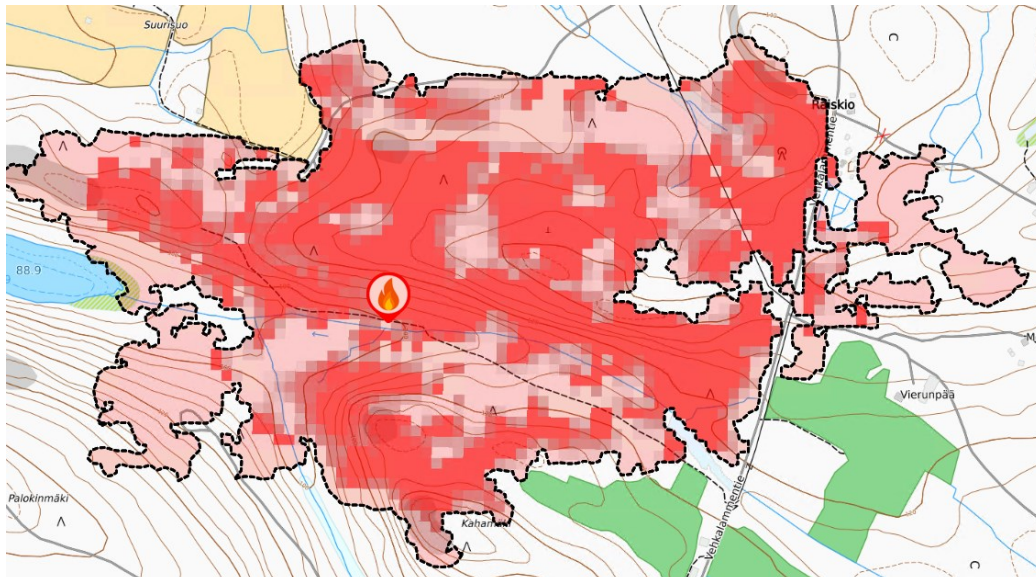
Palon intensiteetti-kartta kuvaa tulipalon intensiteettiä paloalueella, alueen punaisuus kuvaa intensiteetti-tasoa (ks. kuva 14). Mitä punaisempi alue on, sitä intensiivisempää tulipalo on ollut.



Kuva 14 - Prometheus: palon intensiteetti

Tämä taso antaa käyttäjälle kuvan siitä, missä palo on ollut intensiivisimmillään simulaation aikana. Tätä tietoa voidaan hyödyntää esimerkiksi palokatkojen suunnittelussa, jotta palo ei ehdi leviämään intensiivisimmille paloalueille sekä sammutusoperaatioiden turvallisuuden varmistuksessa.

Liekin pituus-karttataso kuuluu myös simulaation tuottamiin kartta-aineistoihin, ja sitä voidaan tarkastella tuntikohtaisesti. Karttatasolta voidaan tarkastella missä liekin korkeus on ollut korkea ja missä matala. Liekin korkeus näkyy kartalla (ks. kuva 15) punaisella: mitä punaisempi väri sitä korkeampi liekki (0–1 m asteikolla). Tätä tietoa voidaan hyödyntää strategisesta näkökulmasta, jonka perusteella voidaan rajata pahimpia paloalueita.

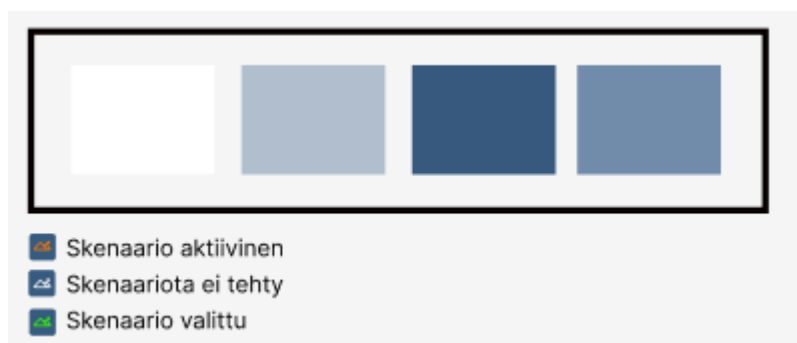


Kuva 15 - Prometheus: Liekin pituus karttataso

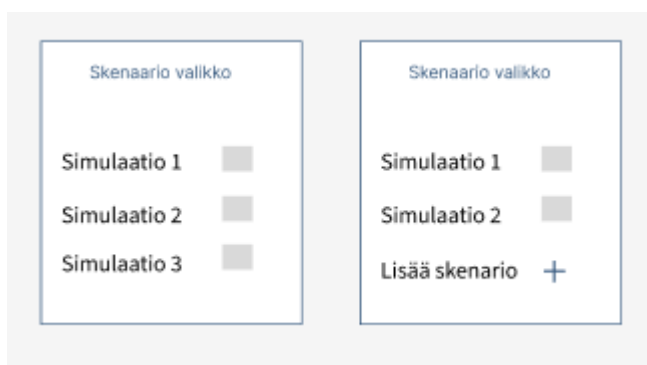
7 TYÖKALUN TOTEUTUS

7.1 Työkalun suunnittelu

Suunnitteluvaiheessa työkalusta laadittiin esityslista PowerPointilla, joka käsiteltiin yhteisissä palaverissa lävitse. Työkalun suunnitteluun käytettiin Figma-alustaa, jossa visuaalinen suunnittelu tapahtui. Tämä sisälsi väriteemat, valikot sekä skenaariotyökalun nappien eri tilat (ks. kuva 16 ja 17). Suunnittelussa pyrittiin vastaamaan ProMS teemaan. Figma on suunnittelualusta erilaisten websuunnittelujen tai projektien pohjalle. Figmalla pystyy jakamaan ja suunnittelemaan käyttöliittymiä sekä toiminnallisuuksia. (Figma 2024.)



Kuva 16 - Figma, teemaluonnos ja skenaarion tilat



Kuva 17 - Figma valikkoluonnos

Tämän laajennoksen taustalla on skenaarioiden rakentamisen sujuvoittaminen, joka saavutetaan käyttämällä yhtä työkalua skenaarioihin liittyvien muuttujien hallinnassa. Työkalussa on 3 eri tilannetta, tilanne 1: skenaariota ei tehty, tilanne 2: skenaario aktiivinen, tilanne 3: skenaario valittu. Näiden tehtävänä on kertoa käyttäjälle skenaarioiden tilanne ja vaihe. Valikosta valitaan skenaario, jota halutaan muokata ja jolle polttoainemuutokset ohjataan. Tämä myös toimii simulaation jälkeen valikkona, josta haluttu skenaario voitaisiin aktivoida, jonka jälkeen skenaariotyökalun tilaa muutettaisiin: skenaario valittu tilaan (ks. kuva 16).

7.2 Työkalun toteutus

Simulaatiotyökalu on toteutettu paikalliseen kehityshaaraan, ja se toteuttaa komponenttiratkaisua. Tämä komponenttiratkaisu määrittää skenaarion valinnan ennen polttoainemuutosten lisäämistä automattisesti. Nykyisessä ratkaisussa jokaisen luodun polttoainemuutoksen skenaario pitää valita erikseen lisäyksen yhteydessä.

Laajennoksen toiminnallisuus on toteutettu Reactilla, joka hyödyntää Reduxin tilanhallintaominaisuutta eli Redux storea. Tilat saadaan käyttöön ProMS:ssa olevilta karttatyökaluilta, joissa nykyinen kartalta valittu kohde on tilanhallinnassa. Komponentti hakee storesta tarvittavat tilat, joiden avulla saadaan kartalta valittu kohde komponentin käytettäväksi. Työkalu hyödyntää valittua kohdetta taustatietona siitä, mitä skenaarioita simulaatiopisteeseen liittyy, jotta näitä voidaan tarkastella työkalun avulla. Reduxin tehtävä on päivittää applikaation näkymää ja Reactin tehtäväksi jää toiminnallisuuden rakentaminen komponenttiin sekä Reduxin tilan päivittäminen tarvittaessa toiminnoilla. Esimerkiksi työkalulla valitaan listalta skenaario, mikä päivittää Redux toiminnallisuutta hyödyntäen tilan muutoksen eteenpäin.

Kun valmiita simulaatioita valitaan kartalta, työkalulla pystytään tarkastelemaan simulaatiopisteeseen liittyviä skenaarioita. Valitusta simulaatiopisteestä avautuu listaan olemassa olevat skenaariot kohteelle, joista voidaan valita haluttu tarkasteltava skenaario simulaatiolle. Jos kartalta ei ole valittu mitään simulaation aloituspistettä, työkalu on "rakennus"-tilassa, eli sen avulla voidaan valita skenaario, johon halutaan lisätä polttoainemuutoksia. Jos valittu piste on simulaatiopiste, työkalu on "tarkastelu"-tilassa, ja valitusta pisteestä voidaan valita olemassa olevia skenaarioita.


Esimerkki tarkastelutilasta: Kartalta valitaan simulaation aloituspiste ProMS:n karttatyökalujen avulla. Tämä tieto päivittyy Reduxin tilanhallintaan, josta se siirtyy komponentille. Kuvassa näkyy skenaario 2 ja simulaatiopisteen ID 243 (ks. kuva 18). Tämän tiedon pohjalta työkalulle haetaan pisteeseen liittyvät skenaariot listalle, joista käyttäjä voi valita haluamansa skenaarion. Tämän jälkeen skenaarioon liittyvät polttoainemuutokset ja simulaatio haetaan näkyville sekä muut ominaisuudet piilotetaan näkyvistä. Kun skenaario vaihdetaan, aiemman skenaarion tiedot piilotetaan kartalta.



Kuva 18 - Skenaarion valinta työkalulla

Esimerkki rakennustilasta: Listalta valitaan skenaario 1, jonka jälkeen voidaan lisätä kartalle haluttu määrä polttoainemuutoksia. Skenaariota ei tarvitse määritellä jokaisen samalle skenaariolle kohdistuvan polttoainemuutoksen lisäyksen yhteydessä. Kuvassa esimerkkinä polttoaineluokan lisääminen ja

skenaarion valinta ilman laajennosta (ks. kuva 19). Kun skenaariot ovat rakennettu, simulaatiopyynnön luominen tapahtuu normaalisti. Tämän jälkeen luotu simulaatio on myös tarkasteltavissa työkalulla tarkastelutilassa.



Kuva 19 - Polttoainemuutoksen valikko

7.3 Työkalun testaus

Työkalun testaukseen käytin selaimen konsolia sekä projektin Redux-logger laajennosta, jonka avulla tilapäivitykset näkyvät selaimen konsolissa (ks. kuva 20). Kuvan esimerkissä nähdään dialogi toiminnallisuuden tilanhallinta. Logger näyttää edellisen tilan ennen toimintaa, toiminnallisuuden, joka muuttaa tilaa sekä lopuksi päivittyneen tilan. Lisäsin myös paikalliseen versioon paljon termi-
naali kommentointia, jotta sain paremman käsityksen tiedon kulusta. Testaus oli suhteellisen moni-
mutkaista, koska projektipohja on hyvin laaja ja tilamuuttujia oli todella paljon.

```

action SET_ALERT_DIALOG @ 12:21:09.759      AlertDialog.jsx:72
  prev state                                AlertDialog.jsx:72
  ▶ {authentication: {...}, tables: {...}, alert: {...}, other: {...}, jun
    ction: {...}, ...}
  action                                    AlertDialog.jsx:72
  ▶ {type: 'SET_ALERT_DIALOG', payload: {...}}
  next state                                AlertDialog.jsx:72
  ▶ {authentication: {...}, tables: {...}, alert: {...}, other: {...}, jun
    ction: {...}, ...}

```

Kuva 20 - Redux-logger

8 POHDINTA

Opinnäytetyön tarkoituksena oli tutustua ja perehtyä ProMS järjestelmän MAST 2-hankkeen simulaatio ympäristöön sekä toteuttaa laajennossuunnitelma sekä implementaatio simulaatiotyökaluun. Työkalua ei saatu toteutettua nykyisessä tilassa vielä järjestelmään, sillä vastaan tuli järjestelmän rajoitteet. Nykyisillä ratkaisulla työn toteuttamiseen menisi liikaa aikaa. Työssä opin tärkeitä komponentteja ProMS:n arkkitehtuurista, kuten Reactista ja Reduxista. Opiskelin työhön Redux käytänteet, jotta ymmärsin kehitysympäristön toimintaperiaatteen paremmin.

Lisäksi opettelin ProMS:n määrittelyn periaatteet ja miten komponentit rakentuvat projektikohtaisesti järjestelmässä tietokannan kautta. Järjestelmän ymmärtäminen tarkoittaa sitä, että ymmärtää, miten määitykset tulevat järjestelmään tietokannasta, miten komponentit rakennetaan käyttöliittymään ja miten tilanhallinta toteutuu. Nämä asettavat tiettyjä ehtoja uusien toiminnallisuuksien lisäämiselle, sillä kaikki projektit toteuttavat ProMS:n työkaluja tarpeen mukaan. Uuden komponentin lisääminen ei ole projektirakenteen näkökulmasta järkevää, jos se ei ole uudelleenkäytettävissä. Tästä syystä skenaario toiminnallisuuteen ei kannata rakentaa omaa komponenttia. Varsinkin, kun kyseessä on olemassa olevaan toiminnallisuuteen tehtävä laajennos. Näihin tapauksiin ProMS tarjoaa projektista riippumattomat ydinkomponentit, joita hyödyntäen toiminnallisuus tulisi tehdä määrittämistietokannan avulla.

Järjestelmä toteuttaa projekteja tietokantojen kautta, eli itse ProMS ei sisällä kuin komponentit toiminnallisuuksien luomiseen tietokannasta. Kaikki ProMS:n komponentit ovat määriteltävissä tietokannan kautta. Tämän vuoksi komponenttipohjainen ratkaisu ei ole ProMS:in ympäristöön sopiva sellaisenaan, vaan vaatii muutostyötä, jotta se voitaisiin toteuttaa tuotantoon.

Työ toteutettiin paikalliseen kehitysversioon esimerkkimallisena ratkaisuna, joka vaatii vielä viimeistelyn sekä rakenteellisen muutoksen, jotta se olisi yhteensopiva ProMS-mallin kanssa. Jos työkalu halutaan lisätä järjestelmään myöhemmin, työkalu pitää rakentaa ProMS:n käytänteitä noudattaen uudelleen. Tämä tarkoittaisi käytännössä sitä, että komponentti pitäisi purkaa osiin, jotka olisivat toteutettavissa tietokannan kautta.

8.1 Työn suunnittelu

Suunnitteluvaiheessa en ollut vielä perehtynyt ProMS:n käytänteisiin ja aloitinkin ympäristöön tutustumisen suunnittelun jälkeen. Tämä aiheutti sen, ettei suunnitelma vastannut kehitysympäristön rajoitteita ja kehityksessä ei huomioitu komponenttiratkaisun haasteita. Tämän takia toteutus eli implementointivaihe jäi kesken. Jos olisin opiskellut kehitysympäristön ennen työkalun suunnittelua, suunnitelmassa olisi osattu huomioida rajoitteet paremmin.

Minulla ei ollut aiempaa kokemusta ProMS kehitysympäristöstä sekä minulle tuli uutena aiheena React Redux ympäristö, joten suunnittelussa olisi kannattanut tehdä asiat päinvastaisessa järjestyksessä kuin nyt toteutettiin. ProMS kehitysympäristön opiskelu vei aikaa, ja suurin osa opinnäytetyön

ajasta menikin kehitysympäristön opetteluun. Reduxia käyttävä React Redux ympäristö on laaja ja moniosainen, ja kokemattomana siinä toimiminen oli haastavaa. Kun ymmärtää Reduxin toiminnan React Redux ympäristössä, työskenteleminen on huomattavasti sujuvampaa ja nopeampaa.

8.2 Simulaatioympäristö

Myös simulaatioympäristöön tutustuminen vei paljon aikaa, sillä simulaatioympäristössä on monia eri työkaluja ja niiden vaikutukset simulaatioihin eroavat. Projektiympäristössä on tärkeä ymmärtää mitkä työkaluista vaikuttavat simulaatiopyyntöihin ja miten simulaatiopyynnot rakentuvat. Polttoainemuutokset ovat pääosassa simulaatiopyyntöjä ja niiden ymmärtäminen on oleellista MAST-2 hankkeen simulaatiopyyntöjen käsittelyssä.

Simulaatioympäristössä työskennellessä on tärkeä ymmärtää, että kyseessä on simulaatio, ei todellinen tapahtuma ja mallien luotettavuus ei vastaa välttämättä aina todellisuutta. Vaikka hyödynnettävät mallit ovatkin matemaattisesti hyvin tarkkoja, simulaatiot ovat aina vain ennusteita ja aputyökaluja. On kuitenkin tärkeää, että näitä työkaluja kehitetään jatkuvasti, jotta ne tulevaisuudessa vastaavat tarkemmin todellisuutta ja ovat luotettavampia sekä tehokkaampia aputyökaluja palontorjunnan suunnittelussa. Simulaatioiden hyödyntäminen lisää myös huomattavasti työturvallisuutta.

LÄHTEET

Arbonaut 2024. Yrityksemme. Verkkojulkaisu. Arbonaut Oy. N.d. <https://www.arbonaut.com/fi/about-us/our-company>. Viitattu 10.3.2024.

Chiarelli, Andrea. Beginning React: Simplify Your Frontend Development Workflow and Enhance the User Experience of Your Applications with React, Packt Publishing, Limited, 2018. E-kirja. Osoitteessa: <http://ebookcentral.proquest.com/lib/savoniafi/detail.action?docID=5477665>. Viitattu 10.3.2024.

Figma 2024. What is Figma. Verkkojulkaisu. Figma. N.d. <https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>. Viitattu 20.4.2024.

Geeks for Geeks 2024. ReactJS State. Verkkojulkaisu. [geeksforgeeks.org](https://www.geeksforgeeks.org) Sanchhaya Education Private Limited. Päivitetty 14.3.2024. <https://www.geeksforgeeks.org/reactjs-state/>. Viitattu 15.4.2024.

Redux 2024. Redux Essentials, Part 1: Redux Overview and Concepts. Verkkojulkaisu. Dan Abramov and the Redux documentation authors. Päivitetty 25.11.2023. <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>. Viitattu 16.3.2024.

Redux 2024. Redux Fundamentals, Part 2: Concepts and Data Flow. Verkkojulkaisu. Dan Abramov and the Redux documentation authors. Päivitetty 30.4.2023. <https://redux.js.org/tutorials/fundamentals/part-2-concepts-data-flow>. Viitattu 15.3.2024.

Tymstra, C.; Bryce, R.W.; Wotton, B.M.; Taylor, S.W.; Armitage, O.B. 2010. Development and Structure of Prometheus: the Canadian Wildland Fire Growth Simulation Model. Pdf-tiedosto. Julkaistu N/D. https://firegrowthmodel.ca/pages/prometheus_documentation_e.html. Viitattu 10.4.2024.