

Koneoppimista hyödyntävä ohjelmistorobottikapasiteetin optimointi

Essi Nevalainen



Tekijä(t) Essi Nevalainen	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Koneoppimista hyödyntävä ohjelmistorobottikapasiteetin optimointi	Sivu- ja liitesivumäärä 41 + 3
<p>Yksi automaation ja robotiikan kehityksen ilmentymä on palveluautomaatio, jota voidaan toteuttaa esimerkiksi ohjelmistorobotiikalla, koneoppimisella ja tekoälyllä. Ohjelmistorobotiikan tullessa yhä tiiviimmäksi ja yleisemmäksi osaksi tietoteknisiä ratkaisuja, tulee käytössä olevien ohjelmistorobottien määrä kasvamaan lähivuosina huomattavasti.</p> <p>Ohjelmistorobotiikan yksi merkittävä kustannus tällä hetkellä ovat lisenssimaksut. Ohjelmistorobotiikan käyttäjille onkin oleellista kyetä hyödyntämään tehokkaasti hankkimiaan lisenssejä, eli toisin sanoen hyödyntää ohjelmistorobottien kapasiteetti optimaalisella tavalla. Yksi kapasiteetin optimointikeinoista on ohjelmistorobottityöntekijöiden tehokas työvuorosunnittelu.</p> <p>Opinnäytetyön tavoitteena on selvittää, voidaanko ennustavaa koneoppimista hyödyntää ohjelmistorobotiikan resurssien optimoinnissa. Lisäksi selvitetään, miten ohjelmistorobottiresurssien optimointia voidaan automatisoida. Työn ulkopuolelle on rajattu ohjelmistorobotin työtehtävien automaattisen sijoittelun tekevän työkalun toteuttaminen.</p> <p>Toiminnallisen opinnäytetyön tuotoksena syntyy ohjelmistorobottikapasiteetin käyttö- ja varustarpeiden kuvaus, tehtäväluettelon malli sekä käyttö- ja varauskalenterin malli.</p> <p>Työn teoriaosuudessa käsitellään ohjelmistorobotiikkaa teknologiana ja esitellään lyhyesti neljä ohjelmistorobotiikkatuotetta. Tämän jälkeen kuvataan kolme tunnistettua lisensointimalia. Lisäksi kerrotaan ohjelmistorobotiikan käyttöönoton hyvistä käytänteistä, sen nykytilasta ja tulevaisuudesta.</p> <p>Resurssien optimointia käsitellään ohjelmistorobottikapasiteetin näkökulmasta. Lisäksi vertaillaan ihmis- ja ohjelmistorobottiresurssien optimoinnin eroja. Tietoperusta päättyy koneoppimista käsittelevään lukuun, jossa avataan sen teoriaa, mahdollisuuksia ja esitellään Microsoft Azure Machine Learning -koneoppimisympäristö.</p> <p>Opinnäytetyö on toteutettu toimeksiantona MOST Digital Oy:lle kevään 2017 aikana. Menetelmänä ovat kirjallisuuskatsaus, asiantuntijahaastattelut sekä työpajamainen innovointi.</p> <p>Opinnäytetyön johtopäätöksenä on, että koneoppimista voidaan hyödyntää ohjelmistorobottikapasiteetin optimoinnissa. Lopuksi annetaan kehitysehdotuksia ja pohditaan tulosten ajankohtaisuutta ja tarpeellisuutta.</p>	
Asiasanat Ohjelmistorobotiikka, koneoppiminen, resurssien optimointi, RPA, Microsoft Azure Machine Learning	

Sisällys

1	Johdanto	1
2	Ohjelmistorobotiikka	4
2.1	Ohjelmistorobotiikkatuotteet	6
2.2	Lisensointimallit.....	8
2.3	Ohjelmistorobotiikan onnistunut käyttöönotto	9
2.4	Ohjelmistorobotiikan nykytilanne ja tulevaisuus.....	10
3	Resurssien optimointi	11
3.1	Ohjelmistorobottikapasiteetin optimointi	11
3.2	Eroavaisuudet ihmis- ja ohjelmistorobottiresurssien optimoinnissa	12
4	Koneoppiminen	14
4.1	Kuinka koneoppiminen toimii.....	15
4.2	Käyttömahdollisuudet ohjelmistorobottikapasiteetin optimoinnissa.....	16
4.3	Microsoft Azure Machine Learning	17
4.3.1	Huomioita tuotteesta	17
4.3.2	Esimerkki prosessiketjusta	18
5	Ohjelmistorobotin työtehtävät.....	21
5.1	Ohjelmistorobotin työtehtävien parametrit	22
5.2	Pakolliset parametrit.....	23
5.3	Lisäarvoa tuottavat parametrit.....	23
5.4	Työtehtävien optimointi	26
5.5	Parametrien jatkokehitys ja ylläpito	29
6	Ohjelmistorobottikapasiteetin käyttö- ja varauskalenterin malli	31
6.1	Tehtävien vienti kalenteriin tehtäväluettelon avulla.....	31
6.2	Kalenterimalli	32
6.3	Kalenterin ja tehtäväluettelon jatkokehitys sekä ylläpito	33
7	Pohdinta.....	34
7.1	Lopputuloksen arvioiminen ja johtopäätökset.....	34
7.2	Jatkokehitystehtävät	35
7.3	Projektin kulku ja erityispiirteet	36
7.4	Tulosten ajankohtaisuus ja tarpeellisuus.....	36
7.5	Oppiminen ja ammatillinen kehittyminen	37
	Lähteet	38
	Liitteet.....	42
	Liite 1. Tehtäväluettelon malli tyhjänä ja esimerkkisisällöllä.....	42
	Liite 2. Kalenterimalli tyhjänä.....	43
	Liite 3. Kalenterimalli esimerkkisisällöllä	44

1 Johdanto

Automaation, digitalisaation ja robotiikan vaikutuksista yhteiskuntaan on kirjoitettu viime vuosina uhkaavia otsikoita. On arvioitu, että 47 % Yhdysvaltojen työntekijöistä työskentelee ammateissa, jotka voidaan siirtää tietokoneiden ja algoritmien tehtäväksi seuraavien 10 – 20 vuoden kuluessa (Frey & Osborne 2013, 38). Suomenkin riskiammattien osuus on tutkimuksessa ollut n. 35 % (Pajarinen & Rouvinen 2014, 3).

OECD:n uudempi tutkimus (Arntz, Gregory & Zierahn 2016, 12–18), jonka lähtökohtana ovat ammatin sijaan siihen kuuluvat erilaiset työtehtävät, antaa lohdullisemman tulevaisuudenkuvan. Tutkimuksen mukaan Suomessa 7 % työvoimasta työskentelee korkean automaation tehtävissä, eli työstä on automatisoitavissa vähintään 70 %. Tulos on OECD-maista kolmanneksi pienin.

Suomen riskiä pienentävät kommunikaatiotaitojen tarpeellisuus työtehtävissä, aikaisempi panostus ICT-kehitykseen ja automaatioon sekä korkea koulutusaste. (Arntz, Gregory & Zierahn 2016, 12–18.) Maailmalla uhkakuvista huolimatta automaatio ja robotiikka ovat etenemässä jättiharppauksin Suomessa ja maailmalla, ja niiden vaikutukset liiketoimintaan ja yhteiskuntaan tulevat olemaan epäilemättä merkittäviä.

Yksi kehityksen ilmentymä on palveluautomaatio, joka tarkoittaa palveluprosessien automatisoimista liiketoimintanäkökulmasta katsottuna (Säisä 21.4.2017). Palveluautomaatiota voidaan toteuttaa esimerkiksi ohjelmistorobotiikalla, koneoppimisella ja tekoälyllä (Lacity & Willcocks 2016a, 33). Näiden teknologioiden tarjoamat kustannussäästöt sekä toiminnan tehostamislupaukset ovat innostaneet yrityksiä ja julkishallintoa kokeilemaan niitä omilla prosesseissaan.

Ohjelmistorobotiikka (robotic process automation, RPA) tarkoittaa tuotteistettua tietojärjestelmien, mukaan lukien käyttöliittymien, erilaisten toimintaketjujen automatisointia. Ohjelmistorobotiikkaa voidaan käyttää välineenä tehokkaaseen palveluautomaatioon sekä IT-tehtävien ja tietojärjestelmien välisen toiminnan automatisointiin. (Säisä 21.4.2017.)

Koneoppiminen (machine learning) taas on datatieteen (data science) tekniikka, joka auttaa tietokoneita oppimaan ennustamaan tulevaisuuden käyttäytymistä, lopputulemia ja kehityssuuntia olemassa olevasta datasta. Koneoppimisessa käytetään matemaattisia kaavoja eli algoritmeja analysoinnissa. (Microsoft 2017a.)

Ohjelmistorobotiikan tullessa yhä tiiviimmäksi ja yleisemmäksi osaksi tietoteknisiä ratkaisuja, tulee käytössä olevien ohjelmistorobottien määrä kasvamaan lähivuosina huomattavasti. Tällöin huomioon on otettava entistä tarkemmin niistä muodostuvat kustannukset.

Ohjelmistorobotiikan yksi merkittävä kustannus tällä hetkellä ovat lisenssimaksut. Joissain tuotteissa pelkkien lisenssien alkuinvestointi on jopa kuusinumeroinen. Tämä johtuu siitä, että joitakin tuotteita ei ole mahdollista ostaa ohjelmistorobotti kerrallaan, vaan hankinta on tehtävä esimerkiksi usean robotin pakettina. Toinen merkittävä kustannus on ohjelmistorobottien toteutustyö, joka pitää sisällään sekä määrittely- että toteutustyötä. (Säisä 21.4.2017.)

Ohjelmistorobotiikan käyttäjille onkin oleellista kyetä hyödyntämään tehokkaasti hankkimaan lisenssejä, eli toisin sanoen hyödyntää ohjelmistorobottien kapasiteetti optimaalisella tavalla. Yksi kapasiteetin optimointikeinoista on ohjelmistorobottityöntekijöiden tehokas työvuorosuunnittelu. Koska ohjelmistorobotteja hallinnoidaan tavanomaisten resurssihallinnan menetelmin, on toimeksiantona toteutetun opinnäytetyön pohjana oletus siitä, että näitä menetelmiä voidaan käyttää myös kapasiteetin optimointiin.

Opinnäytetyöllä tavoitellaan ymmärrystä siitä, voidaanko ennustavaa koneoppimista hyödyntää ohjelmistorobotiikan resurssien optimoinnissa. Ymmärrys kasvaa tarkastelemalla opinnäytetyön tietoperustaa, pohdintaa ja johtopäätöksiä sekä hyödyntämällä projektin lopputuloksia työkaluina asiakastyössä. Projektin lopputuloksena valmistuu ohjelmistorobottikapasiteetin käyttö- ja varaustarpeiden kuvaus, tehtäväluettelon malli sekä käyttö- ja varauskalenterin malli.

Opinnäytetyön tehtävänasetteluna oli selvittää, miten ohjelmistorobottiresurssien optimointia voidaan automatisoida. Vastausta selvitettiin miettimällä, miten ohjelmistorobottien tehtäviä voidaan kirjata ylös ja sijoitella kalenteriin manuaalisesti. Koska koneoppiminen tarjoaa automatisoidun itsestään kehittyvän keinon käsitellä hetkessä suuria datamassoja, todettiin opinnäytetyössä yhdeksi sen käyttösovellukseksi ohjelmistorobotin työkalenterin täyttäminen.

Projekti aloitettiin teoriapohjan kirjoittamisella. Opinnäytetyön toiminnallinen osa aloitettiin kuvaamalla ohjelmistorobottikapasiteetin käyttö- ja varaustarpeet. Tämän jälkeen luotiin käyttö- ja varauskalenterin malli sekä tehtäväluettelo, jolla tehtävät viedään kalenteriin. Projektin tehtävänä ei ollut toteuttaa ohjelmistorobotin työtehtävien automaattista sijoittelutyökalua.

Opinnäytetyön aluksi käsitellään ohjelmistorobotiikkaa, resurssien optimointia ja koneoppimista tutkimuksiin, asiantuntijalähteisiin ja opinnäytetyön tekijän tietämykseen perustuen. Opinnäytetyön tekijä on työskennellyt toimeksiantajalla syksystä 2016 alkaen ja suorittanut ohjelmistorobotiikkatuotteen peruskurssin.

Puolistrukturoituja asiantuntijahaastatteluita toteutettiin kaksi, joista toinen oli parihaastattelu. Haastatteluiden tavoitteena oli hankkia tietoa alueilta, joita ei käsitelty kirjallisissa lähteissä. Haastateltavat toimivat asiantuntijatehtävissä ohjelmistorobotiikkaan erikoistuneissa yrityksissä.

Teorian jälkeen opinnäytetyössä kuvataan projektin tuloksia ja kehitysprosessia. Lopuksi on pohdinta, johon kuuluvat niin projektin johtopäätökset kuin kehitysehdotukset. Lisäksi käsitellään tulosten ajankohtaisuutta ja tarpeellisuutta, sekä opinnäytetyön tekijän oppimista ja ammatillista kehittymistä projektin aikana.

Projektin toimeksiantajana toimi MOST Digital Oy. Keväällä 2016 perustettu startup-yritys on erikoistunut ohjelmistorobotiikan konsultointiin. Lisäksi MOST Digital Oy auttaa asiakkaitaan prosessien ja toimintamallien uudistamisessa, ICT-ratkaisujen kehittämisessä sekä toiminnan digitalisoinnissa (MOST Digital Oy 2016).

Toimeksiantaja on toimittanut ohjelmistorobotiikan ratkaisuja yksityis- ja julkissektorille muun muassa taloushallinnon prosesseihin. Opinnäytetyöllä MOST Digital Oy tavoittelee ymmärrystä ohjelmistorobottikapasiteetin automaattisesta optimoinnista.

2 Ohjelmistorobotiikka

Ohjelmistorobotiikka on tuotteistettua tietojärjestelmien ja käyttöliittymien erilaisten toimintaketjujen automatisointia. Ohjelmistorobotteja voidaan käyttää tietotyön automatisointiin ja kertaluonteisten tai toistuvien tukitoimintojen toteuttamiseen. Ohjelmistorobotille sopivia tehtäviä ovat esimerkiksi tietojen siirrot, muokkaukset ja syöttämiset. (Säisä 21.4.2017.)

Ohjelmistorobotiikkaa käytetään tehokkaan palveluautomaation välineenä (Säisä 21.4.2017). Robotisoinnin avulla muun muassa minimoidaan inhimillisiä virheitä ja saadaan aikaan kustannus- ja aikasäästöjä. Robotisointi tarkoittaa havaittuun liiketoimintaan kehitettyä ratkaisua, jossa osa toiminnasta robotisoidaan, eli ohjelmistorobotti konfiguroidaan suorittamaan osa prosessista.

Kognitiivisia teknologioita voidaan kartoittaa sen mukaan, kuinka autonomisesti ne toimivat ja mitä kognitiivisia tehtäviä ne suorittavat (ks. taulukko 1). Yleinen kehityssuunta on jäseneltyä tietoa ja päätösten asiayhteyttä tarvitsevista koneista siirtyminen yhä vaikeaselkoisempia asiayhteyksiä selvittäviin koneisiin. Ohjelmistorobotiikka on luokituksessa älykkyystasolla *toistuvien tehtävien automatisointi*. Kehittyneemmälle tasolle kuuluu taas koneoppiminen, joka on tasolla *asiayhteyksien ymmärtäminen ja oppiminen*.

Taulukko 1. Mitä nykypäivän kognitiivinen teknologia pystyy tekemään? (Davenport & Kirby 2016, 23)

Älykkyysden tasot

Tehtävä-tyyppi	Tuki ihmiselle	Toistuvien tehtävien automaatio	Asiayhteyksien ymmärtäminen ja oppiminen	Tietoisuus
Numeroiden analysointi	Liiketoimintatiedon hallinta, tiedon visualisointi, hypoteeseihin perustuva analytiikka	Operatiivinen analytiikka, pisteytys, mallien hallinta	Koneoppiminen , neuroverkot	Ei vielä
Sanojen ja kuvien analysointi	Merkintunnistus, puheentunnistus	Kuvantunnistus, konenäkö	IBM Watson, luonnollisen kielen käsittely	Ei vielä
Digitaalisten tehtävien suoritus	Prosessijohtaminen	Ohjelmistorobotiikka , sääntömoottorit	Ei vielä	Ei vielä
Fyysisten tehtävien suoritus	Laitteiden etäohjaus	Teollisuusrobotiikka, yhteistyörobotiikka	Autonomiset robotit ja ajoneuvot	Ei vielä

Automaation skaalasta suurin osa organisaatioista aloittaa ohjelmistorobotiikalla (Lacity & Willcocks 2016a, 65). Lacity ja Willcocks (2016b, 43) toteavat ohjelmistorobotiikan tuotteiden olevan suunniteltu automatisoimaan sääntöihin perustuvia tehtäviä, joissa käsitellään jäsenneltyä tietoa ja joihin on olemassa yksi oikea vastaus. Transaktioiden määrän tulee olla suuri, sekä sääntöjen hyvin dokumentoituja ja selkeitä (Lacity & Willcocks 2016b, 46).

Ohjelmistorobotiikan käyttöönotto edellyttää, että automatisoitava prosessi on vakaa ja tehtävät ovat toistuvia sekä sääntöihin perustuvia. Usein prosessia on muutettava ja standardisoitava ennen kuin se voidaan siirtää ihmiseltä ohjelmistorobotille (Lacity & Willcocks 2016a, 121).

Ohjelmistorobotiikka ei korvaa täysin sovelluskehitystä, mutta sen avulla voidaan lykätä IT-hankintoja jotka saattavat olla kalliita ja aikaa vieviä. Yksi syy käyttöönottoon voikin olla sovellusintegraatioiden puute, jolloin ohjelmistorobotiikkaa voidaan käyttää tietojärjestelmien välisten rajapintojen muodostamiseen (Säisä 21.4.2017). Jos kohdejärjestelmiin tulee muutoksia, on myös ohjelmistorobottia muutettava.

Palveluautomaatio vaikuttaa perusteellisimmin tukitoimintoihin, kuten IT-osastoon, kirjantpitoon ja henkilöstöhallintoon (Lacity & Willcocks 2016a, 45). Ohjelmistorobotiikan on todettu sopivan muun muassa talous- ja henkilöstöhallinnon prosesseihin (Säisä 21.4.2017).

Ohjelmistorobotiikalla voidaan mahdollistaa joustava sopeutuminen vaihtelevaan työmäärään: uusien ihmistyöntekijöiden rekrytoinneille, vanhojen työtehtävien järjestelyille tai perehdytyksille ei ole tarvetta. Esimerkiksi tuotelanseerauksen tukea varten ohjelmistorobottien määrää voidaan nopeasti kasvattaa ja samaan tapaan myöhemmin taas vähentää normaalille tasolle (Lacity & Willcocks 2016b, 44). Ohjelmistorobottien vähentäminen sujuu myös vaivattomammin kuin ihmistyöntekijöiden – lomautuksia, uudelleenkoulutuksia tai irtisanomisia ei tarvita.

Pilvipalveluiden käyttäminen mahdollistaa myös robottien kopioimisen ja levittämisen, jolloin yhden jo valmiiksi opetetun ohjelmistorobotin kyvyt voidaan siirtää minuuteissa tai jopa sekunneissa toisille roboteille pilvessä (Lacity & Willcocks 2016b, 45). Ohjelmistoroboteille onkin järkevää opettaa mahdollisimman monia tehtäviä, jolloin niiden resurssit ovat koko ajan tehokkaassa käytössä (Lacity & Willcocks 2016a, 125).

Ohjelmistorobotiikka tulee oletettavasti vähentämään liiketoimintaprosessien ulkoistusta. Se vaikuttaa alihankkijoiden hinnoittelumalleihin, arvolupauksiin ja sijaintietuihin (Lacity &

Willcocks 2016a, 261). Tämä voi siirtää töitä tulevaisuudessa takaisin länsimaihin ja omaan organisaatioon.

Työ, jossa tehdään paljon toistuvia tehtäviä eri tietojärjestelmiä käyttäen, voi olla ihmiselle hidasta, virhealtista ja epämotivoivaa. Ohjelmistorobotti ei puudu tietojen siirtelyyn järjestelmästä toiseen, ja se kykenee suorittamaan tehtävät samoja käyttöliittymiä käyttäen kuin ihminenkin.

Tietotyön itseään toistavien ja jäseneltyjen tehtävien siirtyminen pois ihmisiltä, joko ohjelmistorobotiikan, koneoppimisen tai tekoälyn seurauksena, vaikuttaa tulevaisuudessa ihmistyöntekijöiden osaamistarpeisiin. Ihmisiltä tarvitaan luovuutta, ongelmanratkaisutaitoja, arviointikykyä ja tunneälyä ratkaisemaan jäsenelämättömien tehtävien esiin tuomat ennalta arvaamattomat haasteet (Lacity & Willcocks 2016b, 48).

Lacity ja Willcocks (2016b, 48) ennustavat viiden vuoden sisällä työpaikoilla työskentelevän entistä enemmän ihmisiä ja ohjelmistorobotteja rinnatusten. Tuotekehityksen ja pilvipalveluihin siirtymisen myötä ohjelmistorobotit eivät tulevaisuudessa tarvitse paljoakaan alkukokoonpanoa tai ohjeistusta. Robotit poimivat, yhdistelevät ja uudelleenjärjestelevät dataa ihmisille arvioitavaksi ja toimenpiteitä varten. Ihmisten tehtäväksi muodostuu tällöin uusien liiketoimintavaatimusten käsittely, vianetsintä, jäsenelämättömien ongelmien ratkaisu, palvelujen luominen sekä asiakassuhteiden luonti. (Lacity & Willcocks 2016b, 48.)

Tulevaisuudessa automatisoinnin ala kehittyy ratkaisemaan järjestelämätöntä tietoa kognitiivisten automaatiotyökalujen keinoin. Tämä mahdollistaa ohjelmistoroboteille esimerkiksi asiakkaiden lähettämien sähköpostien sisällön selvittämisen. Koska ohjelmistorobotit kykenevät käsittelemään suuria massoja dataa nopeasti ja tarjoamaan ratkaisun lähes välittömästi, ovat kehitysmahdollisuudet esimerkiksi asiakaspalvelulle suuret. Tulevaisuudessa asiakaspalvelija pystyy puhelinkeskustelun aikana pyytämään ohjelmistorobottia louhimaan valtavasta datamassasta vastauksen asiakkaan ongelmaan sekunneissa. (Lacity & Willcocks 2016b, 48–49.)

2.1 Ohjelmistorobotiikkatuotteet

Erilaisia ohjelmistorobotiikkatuotteita on tarjolla useita. Tässä opinnäytetyössä tarkasteltiin aihetta neljän Suomessa käytetyn tuotteen näkökulmasta: Blue Prismin, UiPathin, Kapowin ja SikuliXin.

Markkinoilla olevissa tuotteissa on eroja kyvykkyyksien ja ominaisuuksien suhteen. Jotkin ovat sujuvia ottaa käyttöön ja hallinnoida, osa tarjoaa paremman auditoinnin ja turvalli-

suuden ja jotkin taas käsittelevät jäsentelemätöntä dataa paremmin kuin toiset (Lacity & Willcocks 2016a, 253).

Kaikki tuotteet eivät sovellu optimaaliseksi ratkaisuksi jokaiseen liiketoimintaongelmaan. Tämän vuoksi on olennaista ymmärtää robotisointiprojekteissa eri tuotteiden ominaisuudet ja niille sopivimmat käyttökohteet. Tuotteiden eroavaisuudet kuormanjaossa vaikuttavat myös resurssien optimointiin. Seuraavaksi esitellään lyhyesti neljä tuotetta, joista laajimmin Blue Prismiä sen suuren markkinaosuuden vuoksi.

Vuonna 2001 perustettu **Blue Prism** on yksi maailman johtavista ohjelmistorobotiikan toimittajista. Sen yritysasiakkaisiin kuuluvat muun muassa finanssipalvelukonserni Nordea, teknologiayritys IBM ja kulutustavarayhtiö Procter & Gamble. Yritys toimii lisäksi teknologiakumppanina muun muassa konsultointiyritys Accenturelle, tilintarkastusyriyitys Deloitteille ja suomalaiselle ohjelmistorobotiikkaan erikoistuneelle palveluyhtiö Digital Workforce Services Oy:lle. (Blue Prism 2016.)

Blue Prismin markkinaosuus on kotimaassamme vahva sen ollessa Suomen suurimman ainoastaan ohjelmistorobotiikkaa toimittavan Digital Workforce Services Oy:n yksi keskeisimmistä käytössä olevista ratkaisuista. Blue Prismia voidaan verrata niin sanotusti valmistavan teollisuuden tehdasrobotiikkaan, koska pääosin isot volyymit tuottavat Blue Prismin hyödyt (Apostolakis & Takkula 5.4.2017).

Blue Prism soveltuu parhaiten monimutkaisiin, raskaisiin ja suhteellisen vakaisiin robotisointiprosesseihin. Se on pitkälle kehitetty, ominaisuuksiltaan monipuolinen ja tarjoaa automaattisen lokituksen, minkä vuoksi se soveltuu erityisesti prosesseihin, joissa tarvitaan ohjelmistorobotin seurantatietoja tietosensitiivisyyden ja viranomaisvaatimusten vuoksi. Blue Prism ei sovellu parhaiten kevyisiin tai kertakäyttöisiin robotisointeihin sen verraten kalliin hinnan ja perusteellisuuden vuoksi. (Säisä 21.4.2017.)

Toinen suuri ohjelmistorobotiikan toimittaja **UiPath** perustettiin vuonna 2012. UiPathin kumppaneihin kuuluvat muun muassa konsulttiyhtiö Capgemini, monialakonserni Tech Mahindra ja suomalainen taloushallinnon automatisointiin erikoistunut OpusCapita Group Oy. (UiPath 2017.) UiPath soveltuu parhaiten käyttöliittymässä tapahtuviin nopeisiin robotisointeihin. Se ei sovellu niinkään ylläpidettäviin, pitkäaikaisiin robotisointeihin. (Säisä 21.4.2017.)

Monikansallinen vuonna 1985 perustettu ohjelmistoyritys Kofax on kehittänyt ohjelmistorobotiikkatuotteen **Kapow**. Suomessa sitä käyttää esimerkiksi Festum Software Oy (Fes-

tum 2017). Kapow soveltuu niin nopeisiin, kuin ylläpidettäviin robotisointeihin (Säisä 21.4.2017).

Vuonna 2009 avoimen lähdekoodin opiskelijaprojektina alkanut **SikuliX** eroaa kaupallisista ohjelmistorobotiikkatuotteista ja on lähimpänä perinteisiä testausautomaatiotuotteita (SikuliX 2017). Sitä voidaan käyttää erityisesti nopeisiin ja kevyisiin robotisointeihin (Säisä 21.4.2017).

2.2 Lisensointimallit

Lisenssit muodostavat olennaisen osan ohjelmistorobotiikan kustannuksista. Vaikka Suomessa on tällä hetkellä enemmänkin ylitarjontaa kapasiteetista, ollaan täälläkin menossa samaan suuntaan kuin ulkomailla, jossa ohjelmistorobotteja sekä niiden myötä kustannuksia on enemmän (Apostolakis & Takkula 5.4.2017).

Nykyään tarjolle on tullut myös ilmaisia työkaluja (Tivi 2017). Markkinan kasvaessa hinnat tulevatkin luultavasti nykytasoa alemmas. Opinnäytetyön yhteydessä tunnistettiin kolme erilaista lisensointimallia: avoimeen lähdekoodiin perustuva ilmainen malli, ohjelmistorobottikohtainen malli sekä palvelinkapasiteettikohtainen malli.

Avoimella lähdekoodilla toimivat ohjelmistorobotiikkatuotteet, kuten SikuliX, ovat ilmaisia, jos kehitys- ja ylläpitotyötä ei hanki palveluntarjoajalta. Jos kuitenkin toimitaan työpöytäympäristössä, on optimoinnilla saavutettavissa kustannushyötyjä.

Ohjelmistorobotit voivat käyttää joko työpöytää tai rajapintoja. Jos ohjelmistorobotti toimii työpöytäympäristössä, jolloin se käyttää ohjelmia kuten ihminen, on sille annettava pääsy jonkin työntekijän työpöydälle tai luotava sen käyttöön oma virtuaalinen työpöytä. Koska tehokkaampaa on käyttää virtuaalityöpöytää, syntyy siihen asennettavista ohjelmistolisensseistä kustannuksia. (Säisä 21.4.2017.)

Rajapintojen kautta toimivat ohjelmistorobotit ovat työpöydässä toimivia tehokkaampia, mutta aina niiden tekeminen ei ole mahdollista. Lisäksi osa tehtävistä on kustannustehokkaampia toteuttaa suoraan käyttöliittymässä. Käyttöliittymässä toimivat ohjelmistorobotit toimivat työpöydän kautta, jolloin yksi ohjelmistorobotti voi tehdä vain yhtä tehtävää. Tällöin on perusteltua pyrkiä optimoimaan resurssien käyttö. (Säisä 21.4.2017.)

Ohjelmistorobottikohtaisessa mallissa voidaan ajaa vain yhtä tehtävää kerrallaan. Ohjelmistorobottikohtainen lisensointi perustuu robottiyksiköihin, joille tehtäviä voidaan jakaa kuin ihmiselle. Työaika on 24 tuntia työtä päivässä ympäri vuoden. (Säisiä 21.4.2017.) Esimerkiksi Blue Prism on robottiyksikkökohtainen.

Palvelinkapasiteettikohtaisessa mallissa työtä voidaan teettää niin paljon kuin lisensoitu palvelinkapasiteetti kykenee pyörittämään. Tässä mallissa tehtäviä voidaan ajaa rinnakkain, minkä vuoksi työtehtävien aikataulut on kannattavaa optimoida. (Säisiä 21.4.2017.)

2.3 Ohjelmistorobotiikan onnistunut käyttöönotto

Ohjelmistorobotiikkaprojektien onnistumisen edellytyksenä voidaan pitää vahvaa liiketoimintavetoisuutta. Liiketoiminnan johto kykenee parhaiten tunnistamaan ja priorisoimaan ne prosessit ja tehtävät, jotka tuottavat eniten arvoa asiakkaille ja työntekijöille. IT-osaston osallistaminen ajoissa tarjoaa myös etuja projektille. (Lacity & Willcocks 2016b, 46.)

Organisaation sisäinen kommunikaatio automatisointistrategiasta, aikataulusta ja tulevien muutosten vaikutuksista työntekijöihin on olennaista onnistumiselle (Lacity & Willcocks 2016b, 45). Ajoissa tehty, avoin kommunikointi robotisoinnin aiotuista vaikutuksista työhön on todettu kriittiseksi osaksi prosessia. Riittämätön kommunikaatio saattaa saada työntekijät yliarvioimaan epätoivotut seuraamukset, kuten työtehtävien vähenemisen tai irtisanomiset. Lacity ja Rottman (2008, teoksessa Lacity & Willcocks 2016b, 47) havaitsivat työntekijöiden joissakin tapauksissa jopa sabotoineen aloitteita pelon vuoksi.

Robotisoinnilla tavoiteltujen hyötyjen kuvaaminen asiakkaille ja työntekijöille edistää projektin onnistumista ja innostusta robotisointiin organisaatiossa yleisemminkin. Eräässä uutistoimiston projektissa ohjelmistorobotit opetettiin koostamaan yritysten tulosraportteja. Asiakkaat kokivat positiiviseksi aiempaa laajemman raportoinnin, kun taas uutistoimiston työntekijät pitivät epämieluisan työtehtävän poistumisesta. (Lacity & Willcocks 2016b, 46.)

Jos organisaatio haluaa ohjelmistorobotiikan olevan tulevaisuudessa tiivis osa sen toimintaa, on osaamiskeskuksen perustaminen tarpeen. Osaamiskeskus toimii organisaation jaettuna resurssina, vakiinnuttaa standardit ja parhaat käytännöt sekä seuraa automaatioliiketoiminnan suorituskykyä. Lisäksi osaamiskeskus auttaa yksiköitä kohteiden tunnistamisessa, priorisoi projekteja, rakentaa ratkaisuja sekä valvoo ohjelmistorobotteja. Osaamiskeskus kykenee lisäksi uusiokäyttämään robotteja sekä aiemmista projekteista robottikirjastoon syntyneitä olioita. (Lacity & Willcocks 2016b, 47–48.)

Lacityn ja Willcocksin (2016b, 43) tutkimuksessa havaittiin kauaskantoisesti katsovien yritysten saavuttavan suurimman hyödyn palveluautomaatiosta, verrattuna yrityksiin jotka tavoittelivat sillä vain nopeita hyötyjä. Parhaimmat tulokset saavuttaneilla yrityksillä ei ollut palveluautomaatiostrategiaa itsessään.

Sen sijaan niillä oli strategioita, jotka määrittivät organisaation pitkän aikavälin tavoitteet: esimerkiksi joustavamman henkilöstön luomisen tai palveluiden kasvattamisen ilman työntekijöiden lisäämistä. Nämä strategiat olivat johdon ajamia ja palveluautomaation mahdollistamia – täten palveluautomaatio oli avaintekijä liiketoiminnan muutoksessa. (Lacity & Willcocks 2016b, 43.)

2.4 Ohjelmistorobotiikan nykytilanne ja tulevaisuus

Tällä hetkellä ohjelmistorobotiikka on ottamassa Suomessa ensi askeleitaan, mutta kasvu on suurta. Alan yrityksiin on tehty viime aikoina miljoonasijoituksia (Tivi 2017). Robotisointiprojekteja on ehditty toteuttaa yritysten lisäksi myös julkiselle puolelle: esimerkiksi Espoon kotihoito ja Valtion talous- ja henkilöstöhallinnon palvelukeskus ovat ottaneet ohjelmistorobotteja käyttöönsä (Espoon kaupunki 2017; Palkeet 2016).

Kun ohjelmistorobottien lukumäärä organisaatioissa kasvaa tulevaisuudessa, on lisääntyvien kustannusten vuoksi resurssien optimoinnille aiempaa suurempi tarve (Apostolakis & Takkula 5.4.2017). Tämä antaaakin potentiaalisen opinnäytetyössä kehitettäville optimointimenetelmille.

Suomessa ohjelmistorobotiikkapalveluita tarjoavia yrityksiä on tällä hetkellä arviolta parikymmentä. Joukkoon kuuluu niin ohjelmistorobotiikan konsultointiin erikoistuneita yrityksiä, kuten opinnäytetyön toimeksiantaja MOST Digital Oy, kuin ohjelmistorobotiikkaa toimittavia yrityksiä kuten Digital Workforce Services Oy.

3 Resurssien optimointi

Resurssilla tarkoitetaan voimavaraa, kuten materiaalia tai työvoimaa, josta hyötyä tuotetaan. Opinnäytetyössä tarkoitettuja resursseja ovat ohjelmistorobotit: tuotteesta riippuen robottiyksiköt tai palvelinkapasiteetti. Resurssien hallinnalla varmistetaan ohjelmistorobotien saatavuus oikeaan aikaan sekä niiden riittävyys ja tehokas käyttö (Karlos, Kujala & Martinsuo 2006, 121–122).

Optimoinnilla tarkoitetaan määriteltyjen resurssien käyttöä parhaalla mahdollisella tavalla ja haluttujen tavoitteiden mukaisesti (Niikko 2015, 22). Opinnäytetyössä resurssien optimoinnilla tarkoitetaan tehtävien tekemistä ja niiden sijoitlemistä siten, että

- asiakkaan vaatimukset tehtävien priorisoinnista ja suoritusajankohdasta täyttyvät
- ohjelmistorobotteja kuormitetaan tasaisesti
- ohjelmistorobottien käyttöaste on mahdollisimman korkea.

Jos tehtävien kesken on riippuvuuksia, on ne otettava optimoinnissa huomioon. Esimerkiksi jotta robotti A voi tulostaa raportin järjestelmästä, on robotti C:n pitänyt syöttää uudet työntekijät järjestelmään. Tällöin robotti A:n tehtävä on riippuvainen robotti C:n tehtävän suorituksesta. Riippuvuuksia voidaan kuvata esimerkiksi tehtäväverkolla, joka on tehtävien ja niiden keskinäisten riippuvuuksien graafinen esitystapa (Karlos ym. 2006, 131).

Resurssien tasoittaminen tarkoittaa työkuorman jakamista resurssien kesken niin, että työkuorma on mahdollista hoitaa järkevästi. Niin yli- kuin alikuormitukset pyritään karsimaan pois. (Karlos ym. 2006, 144.)

Onnistuneen resurssien optimoinnin avulla tehtävät tulevat suoritetuiksi ja maksetut lisenssit hyödynnetään mahdollisimman tehokkaasti. Epäonnistunut resurssien optimointi näkyy toiminnassa esimerkiksi siten, että

- asiakkaan vaatimukset tehtävien priorisoinnista ja suoritusajankohdasta eivät täyty
- ohjelmistorobottien kuormitus ei ole tasaista
- ohjelmistorobottien käyttöaste jää matalaksi
- ohjelmistoroboteista on ajoittain puutetta tai ylitarjontaa.

3.1 Ohjelmistorobottikapasiteetin optimointi

Koska ohjelmistorobotteja hallinnoidaan kuten ihmistyötä eli työjonojen kautta, on opinnäytetyön oletuksena se, että näillä menetelmillä voidaan hallinnoida myös ohjelmistorobotteja. Ohjelmistorobotin työn optimointi on vieläpä huomattavasti helpompaa ihmistyöhön verrattuna. Eroavaisuuksia on esitelty seuraavassa alaluvussa tarkemmin.

Resurssien optimoinnin näkökulmasta ohjelmistorobotteja voidaan ajatella myös työkaluina tai työkoneina, joita halutaan käyttää tehokkaasti. Opinnäytetyössä jatkettiin kuitenkin alalla vakiintunutta näkemystä siitä, että ohjelmistorobotit ovat digitaalista työvoimaa ihmisten rinnalla.

Ohjelmistorobottikapasiteetin optimointia vaikeuttaa se, että ohjelmistorobotit ovat melko ympäristökeskeisiä: kun yksi suunnitellaan toimimaan yhdellä koneella, se ei välttämättä toimi toisella koneella ilman viimeistelyä. Ohjelmistorobotteja myös harvemmin täytetään ääriään myöden, koska niiden käyttäjätunnuksille ei haluta turvallisuuskysymysten vuoksi antaa liian laajoja käyttöoikeuksia. Siksi tietyille ohjelmistoroboteille annetaan pääsy tiettyihin järjestelmiin, ja vain osalla ihmisistä on pääsy tiettyihin roboteihin. (Apostolakis & Takkula 5.4.2017.)

Optimointia ei myöskään aina kannata tehdä kovin tiiviiksi, koska asiakas on joskus valmis maksamaan varmuudesta tehokkuuden kustannuksella. Tämä koskee etenkin kriittisiä toimintoja. (Apostolakis & Takkula 5.4.2017.) Prioriteetti-parametrilla voidaan kuvata tehtävän kriittisyyttä liiketoiminnalle, jolloin korkean prioriteetin tehtävät saavat etuoikeuden aikataulutuksessa.

3.2 Eroavaisuudet ihmis- ja ohjelmistorobottiresurssien optimoinnissa

Koska ohjelmistorobotit nähdään laajalti digitaalisina työntekijöinä pelkkien tietokoneohjelmistojen sijaan, ovat ne myös rinnastettavissa ihmistyöntekijöihin. Mitä eroja sitten on havaittavissa näiden kahden erilaisen työvoiman optimoinnissa?

Ohjelmistorobottiresurssien optimointi on ihmisresurssien optimointia huomattavasti suoraviivaisempaa. Kun inhimillinen tekijä poistetaan, ei työvuorojen ketjutusta, kokemustaustaa, jaksamista tai henkilökemioita tarvitse huomioida työvuorosuunnittelussa. Työaikalaki ei rajoita ohjelmistorobottien työpäivän pituutta, eivätkä ne tarvitse vapaapäiviä tai lomakoikoja.

Ohjelmistorobotit tekevät vain sen mitä on suunniteltu ja työskentelevät kyllästymättä, väsymättä tai harhautumatta. Ihmisen kokonaistyöaikaa pienentäviä tehtäviä ja keskeytyksiä ei ohjelmistorobottiresurssien optimoinnissa tarvitse ottaa huomioon. Pelinin (2011, 154) mukaan näitä ovat

- puhelinsoitot
- vierailijat
- tehottomat kokoukset
- ohjelmistojen opettelu
- perehdyttäminen

- myynti
- tarjoukset.

Ohjelmistoroboteille voidaan opettaa usean eri osaamisprofiilin tehtäviä, jolloin työnkuva voi olla laajempi kuin mitä yksittäisellä ihmisellä usein on. Ohjelmistorobotti saattaaakin hallita usean osaston ja työntekijän erilaisia työtehtäviä, ja se pystyy vaihtamaan tehtävien välillä sujuvasti. Osaamisprofiilin joustavuus helpottaa ohjelmistorobottiresurssien tehokasta käyttöä: jos ilmenee tyhjäkäyntiä, voidaan uusia tehtäviä opettaa hyvinkin laajalla skaalalla.

4 Koneoppiminen

Koneoppiminen on tekoälyn osa-alue, jossa tutkitaan oppivien algoritmien ja tekniikoiden kehittämistä. Algoritmi on itsenäinen joukko sääntöjä, jota käytetään ongelmien ratkaisussa tietojenkäsittelyyn, matematiikan tai automaattisen päättelyn kautta (Microsoft 2017a).

Koneoppimisen tavoitteena on opettaa koneita löytämään ratkaisuja ongelmiin datan avulla ja kehittymään automaattisesti tekemään uusista ennusteista tarkempia. Tekoälyn pioneeri Arthur Samuel määritteli vuonna 1959 koneoppimisen olevan tutkimusala, joka antaa tietokoneille kyvyn oppia ilman nimenomaista ohjelmointia (Munoz 2014, 1).

Koneoppiminen on tällä hetkellä muotitekniologia, ja Gartner ennustaakin sen tulevan valtavirran omaksumaksi kahden – viiden vuoden sisällä (Gartner 2016). Esimerkiksi IBM ja Google ovat jo sijoittaneet miljardeja koneoppiviin algoritmeihin (Pauku 2016, 8).

Kyseessä ei kuitenkaan ole uusi teknologia. Matemaattiset ideat koneoppimisen takana ja useat käytetyistä algoritmeista ovat parikymmentä vuotta vanhoja. Viimeaikaiset edistysaskeleet laskentanopeudessa, muistissa, tiedonhaussa, antureissa ja algoritmeissa ovat kuitenkin yhdessä alentaneet dramaattisesti koneoppimiseen pohjautuvien ennusteiden kustannuksia. (Agrawal, Gans & Goldfarb 2017, 24.)

Kustannusten aleneminen on lisännyt koneoppimisen sovelluskohteita. Jos käytät esimerkiksi Googlen tai Netflixin ohjelmia sekä jotain pankkipalvelua, olet jo osa oppivan koneohjelman tietokantaa. Näihin tietokantoihin on koottu tietoa jopa satojen miljoonien ihmisten tavoista ja käyttäytymisestä, minkä vuoksi ohjelmat tuntuvatkin aavistavan, mitä haluat ja teet. Oppivien laskentaohjelmien algoritmit oppivat sitä enemmän, mitä enemmän niitä käyttää. (Pauku 2016, 8–9.)

Tulevaisuudessa koneoppiva ohjelma saattaa automatisoida esimerkiksi uusien lääkkeiden löytämisen tai räätälöidä syöpähoidot. Manchesterin yliopistossa lääkkeitä keksii koneoppimista hyödyntävä robottitutkija Eve, johon on tallennettu molekyylibiologian perusteet ja paljon oheistietoja. Vuonna 2015 Eve löysi itse uuden malarialääkkeen. (Pauku 2016, 8–9.)

4.1 Kuinka koneoppiminen toimii

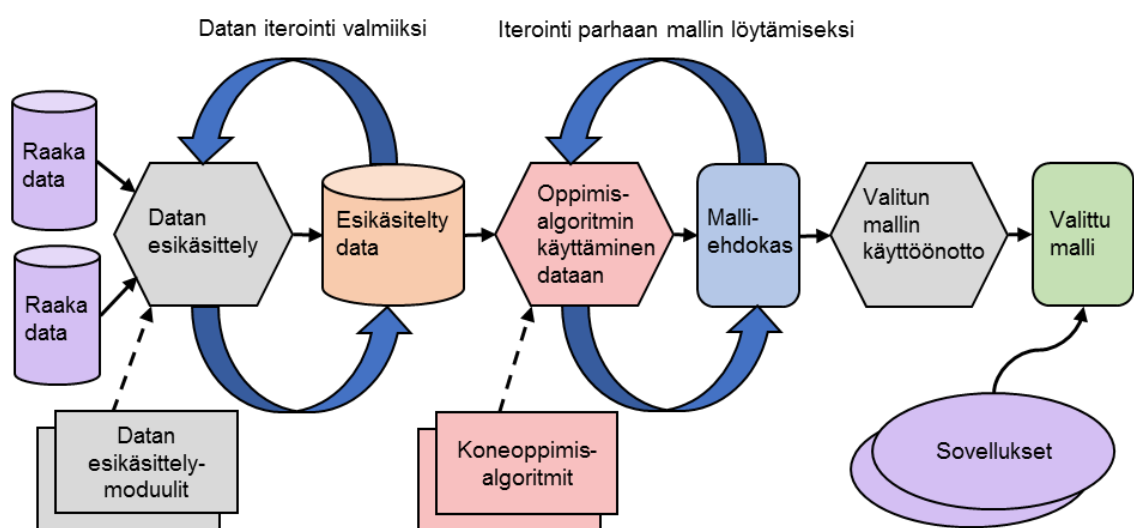
Tässä luvussa käsitellään koneoppimisprosessia ja sitä, miten koneoppiminen toimii. Koneoppimisen voi jakaa kolmeen kategoriaan: ohjattuun oppimiseen, ohjaamattomaan oppimiseen ja vahvistusoppimiseen (Lyytinen 2017, 3–4).

Ohjatussa oppimisessa kone opetetaan esimerkiksi tunnistamaan numeroita näyttämällä sille valtava määrä kuvia ja kertomalla mikä numero kuvassa on. Ohjattuun oppimiseen perustuu esimerkiksi Googlen kehittämä AlphaGo-ohjelma, joka päihitti ensi kertaa ammattilaispelaajan kiinalaisessa strategisessa go-lautapelissä. (Saarinen 2016, 24.)

Rajotteena nykyisessä ohjatussa oppimisessa on se, että kone tarvitsee suuren määrän valmiiksi luokiteltua aineistoa (Saarinen 2016, 24). Se on kuitenkin suosittu ja käytännöllinen koneoppimistapa, ja esimerkiksi Microsoft Azure Machine Learning -koneoppimistuotteessa kaikki paitsi yksi moduuli ovat ohjatun oppimisen algoritmeja (Microsoft 2017b).

Ohjaamaton oppiminen taas alkaa raakadatasta, jota ei tarvitse luokitella tai määritellä etukäteen. Se jäljittelee ihmisen aivokuoren toimintaa. Vahvistusoppimisessa kone taas oppii ympäristön antaman palautteen perusteella. (Lyytinen 2017, 3–4.)

Kuvassa 1 esitellään koneoppimisprosessi Chappellin mukaan (2015, 5–7). Prosessi alkaa käsittelemättömällä datalla, jota on hyvä olla mahdollisimman paljon. Mitä enemmän dataa on, sitä todennäköisemmin siitä johdetut tulokset ovat osuvia.



Kuva 1. Koneoppimisprosessi alkaa käsittelemättömällä datalla ja päättyy tästä datasta johdettuun malliin (Chappell 2015, 5)

Kriittisen tärkeää prosessin onnistumiselle on valita oikeanlaista dataa käsiteltäväksi. Tässä liiketoiminnan asiantuntijoilla on useimmiten paljon annettavaa. Esimerkiksi luottokorttipetoksen tunnistamisessa alan asiantuntijat osaavat kertoa, mitkä mittarit osoittavat todennäköisesti petoksen. Huomioon tulee ehkäpä ottaa yritys, jossa korttia käytettiin, asiakkaan ikä sekä vuorokaudenaika jona korttia käytettiin. (Chappell 2015, 5–6.)

Lähtödata vaatii useimmiten esikäsittelyä, ennen kuin prosessissa voidaan edetä. Luottokorttipetoksen datassa saattaa olla esimerkiksi puuttuvia tietoja tai duplikaatteja kirjauksissa. Esikäsittelyssä käytetään esikäsittelymoduuleita ja iteroidaan, kunnes data on onnistuneesti valmisteltua. Usein oikean datan valitseminen ja esikäsittely vievät valtaosan ajasta koneoppimisprojektissa. (Chappell 2015, 6.)

Kun data on esikäsitelty, siirrytään etsimään parasta tapaa ratkaista ongelma. Prosessissa käytetään koneoppimisalgoritmeja, jotka tyypillisesti soveltavat tilastollista analyysia dataan. Tavoitteena on määrittää, mikä yhdistelmä algoritmia ja esikäsiteltyä dataa aikaansaa käyttökelpoisimman tuloksen: esimerkiksi mikä yhdistelmä ennustaa todennäköisimmin, onko luottokorttitapahtuma petos vai ei. (Chappell 2015, 6.)

Kun prosessissa sovelletaan algoritmia esikäsiteltyyn dataan, tuloksena saadaan malliehdokas. Sopivan mallin löytäminen vaatii useimmiten iteroimista eri yhdistelmiä käyttäen. Malli ei useinkaan anna kyllä/ei-vastausta, vaan palauttaa todennäköisyyden välillä 0 ja 1. Sen määrittely, mikä skaalasta tarkoittaa esimerkiksi todennäköistä petosta ja mikä ei, on liiketoiminnallinen päätös. Punnittavana on muun muassa se, onko liiketoiminnalle tärkeämpää saada petokset kiinni, vaikka sillä riskeerattaisiinkin hyvä asiakaskokemus. (Chappell 2015, 6.)

Malli on koneoppimiskokeen tulos, joka koostuu opetusdatasta, algoritmimoduulista ja funktionaalisista moduuleista kuten pisteytysmoduulista (Microsoft 2017a). Mallin löytäminen päättää prosessissa itse koneoppimisen osuuden. Viimeinen askel on kuitenkin ottaa malli käyttöön. Tämä sallii sovellusten käyttää algoritmia jota malli toteuttaa. Malleja käytetään sovellusten kautta vastaamaan esimerkiksi kysymykseen ”Onko tämä luottokorttitapahtuma petollinen?”. (Chappell 2015, 7.)

4.2 Käyttömahdollisuudet ohjelmistorobottikapasiteetin optimoinnissa

Ohjelmistorobottikapasiteetin optimoinnin näkökulmasta voidaan koneoppimista käyttämällä hyödyntää ainakin kolmea asiaa. Ensimmäinen ja oleellisin on koneoppimisen kyky tehdä ennusteita datan perusteella. Tätä voidaan hyödyntää, kun käytössä on riittävästi

toteumadataa, jonka avulla voidaan ennustaa resurssiipukkeja ja vapautuvia resursseja. Resurssiipukkeilla tarkoitetaan tilanteita, joissa ohjelmistorobotteja tarvitaan ennakoitua enemmän.

Toinen asia jota voidaan hyödyntää, on koneoppimisella saatava joustava ja lähes rajaton laskentakapasiteetti. Sen avulla kyetään käsittelemään suuriakin tietomassoja. Työtehtävien sijoittelua ja ennusteiden tekemistä voidaan suorittaa silloin kun sille on tarvetta – näin voidaan esimerkiksi mukauttaa ohjelmistorobottien aikatauluja reaaliaikaisesti.

Kolmas hyödynnettävä asia on se, että koneoppimispalvelut ovat helposti julkaistavissa www-sovelluspalvelumuodossa. Www-sovelluspalvelut (web service) ovat verkkopalvelimessa toimivia ohjelmia, jotka tarjoavat standardoitujen internetyhteyskäytäntöjen avulla palveluja sovellusten käytettäväksi (Tietotekniikan termitalkoot 2012). Www-sovelluspalvelun avulla voidaan toteuttaa yleiskäyttöisiä komponentteja sekä ohjelmistorobottien tueksi, että resurssien optimoimiseksi.

4.3 Microsoft Azure Machine Learning

Microsoft Azure Machine Learning (Azure ML) on selainpohjainen ennustavan analytiikan pilvipalvelu, jonka avulla on mahdollista luoda ja ottaa käyttöön ennustavia malleja (Microsoft 2017c). Azure ML esitellään tässä luvussa lyhyesti yhtenä esimerkkinä pilvipalveluna toteutetusta koneoppimisympäristöstä.

Opinnäytetyössä ei oteta kantaa siihen, kannattaisiko työtehtävien sijoittelutyökalu toteuttaa juuri Azure ML -tuotteella. Muita koneoppimistuotteita ei kuitenkaan projektin aikana kokeiltu. Microsoftin lisäksi markkinoilla olevia koneoppimistuotteita ovat esimerkiksi Google Prediction API, Amazon Machine Learning ja IBM Watson (InfoWorld 2016).

4.3.1 Huomioita tuotteesta

Azure ML:n eduksi voidaan laskea se, että kyseessä on pitkälle tuotteistettu koneoppimisympäristö. Tarjolla on videoita, ohjeita ja dokumentaatiota harjoittelun tueksi. Käyttäjä voi hyödyntää harjoittelussa myös valmiita prosessiketjuja (experiment) ja esimerkkietoaineistoja (dataset).

Koneoppimisympäristöä voi kokeilla ilmaiseksi ja sen käyttöönotto on nopeaa. Jos organisaatio käyttää jo Microsoftin Azure-pilvipalvelua, on integraatio Azure ML:ään sujuvaa (Louhia 2016; Digital Illustrated 2015). Yleiskäyttöinen optimointipalvelu, jota eri tahot voi-

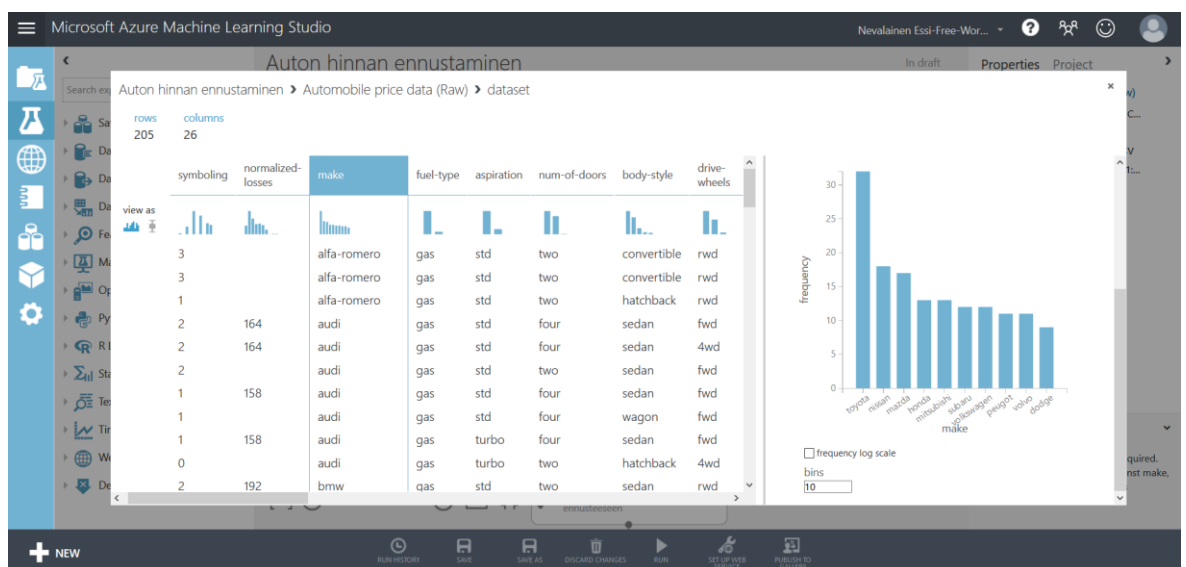
vat käyttää standardoidun rajapinnan kautta, on mahdollista toteuttaa Azure ML:ään www-sovelluspalveluna.

Azure ML sisältää paljon valmiita toimintoja ja algoritmeja ennustamiseen. Azure ML mahdollistaa myös Python- ja R-ohjelmointikielen käyttämisen moduulien personoimisessa. Pythonin etuja ovat yksinkertainen syntaksi, alustariippumattomuus ja laaja kokoelma tehokkaita kirjastoja (Microsoft 2017d). R-kieli on tilastotieteelliseen ohjelmointiin ja grafiikan tekemiseen tarkoitettu ohjelmointikieli, jota tilastotieteilijät ja data-analyytikot käyttävät algoritmien toteuttamisessa (Microsoft 2017e).

Jos tietojen halutaan pysyvän Suomen rajojen sisäpuolella, voi Microsoftin palvelinkeskusten sijoittelu olla joissakin organisaatioissa este Azure ML:n käytölle. Euroopan Azure ML -keskukset sijaitsevat Hollannissa ja Saksassa (Microsoft 2017f). Vaikka Microsoft pyrkii vastaamaan vuonna 2018 voimaantulevaan EU:n tietosuoja-asetukseen ja sen myötä tarkentuvaan sääntelyyn, on pilvipalvelua käyttöönotettaessa oltava selvyyttä omista velvollisuuksista (Microsoft 2017g).

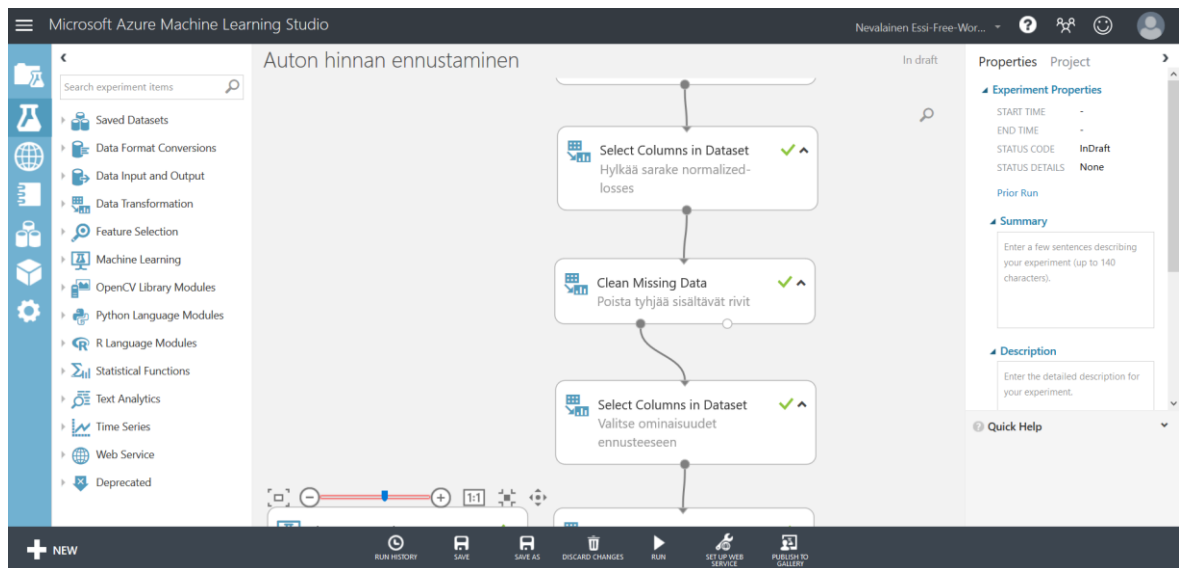
4.3.2 Esimerkki prosessiketjusta

Azure ML:ssä prosessiketjujen tekeminen alkaa datalla. Tässä esimerkissä seurattiin Microsoftin opasta auton hinnan ennustamiseen (Microsoft 2017h). Dataa voidaan visualisoida muutamalla klikkauksella jo tässä vaiheessa. Kuvassa 2 näkyy osa aineiston riveistä ja histogrammi valittuna olevasta valmistaja-sarakkeesta. Suurtakin aineistoa voidaan tarkastella helposti visualisoimalla.



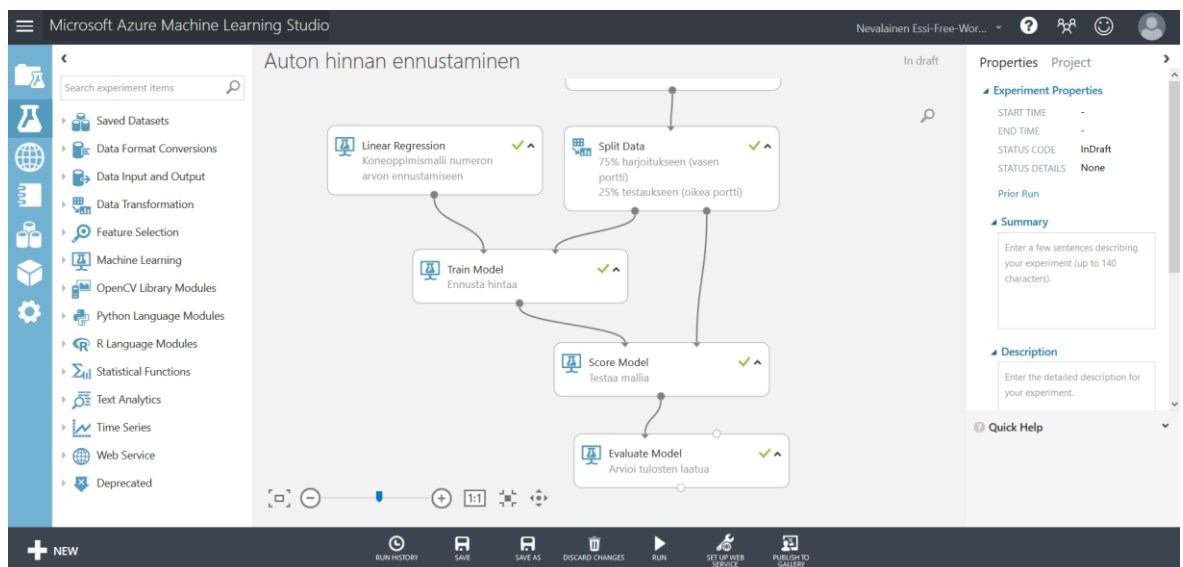
Kuva 2. Visualisointi-ikkunassa voidaan tarkastella dataa

Seuraavaksi data valmistellaan käsiteltäväksi. Tässä tapauksessa puuttuvia arvoja sisältävät rivit poistetaan, jotta malli voi analysoida datan oikein. Myös yksi sarake, jossa puuttuvia tietoja on paljon, poistetaan. Tämän jälkeen valitaan ennusteelle olennaiset ominaisuudet. (Kuva 3.) (Microsoft 2017h.)



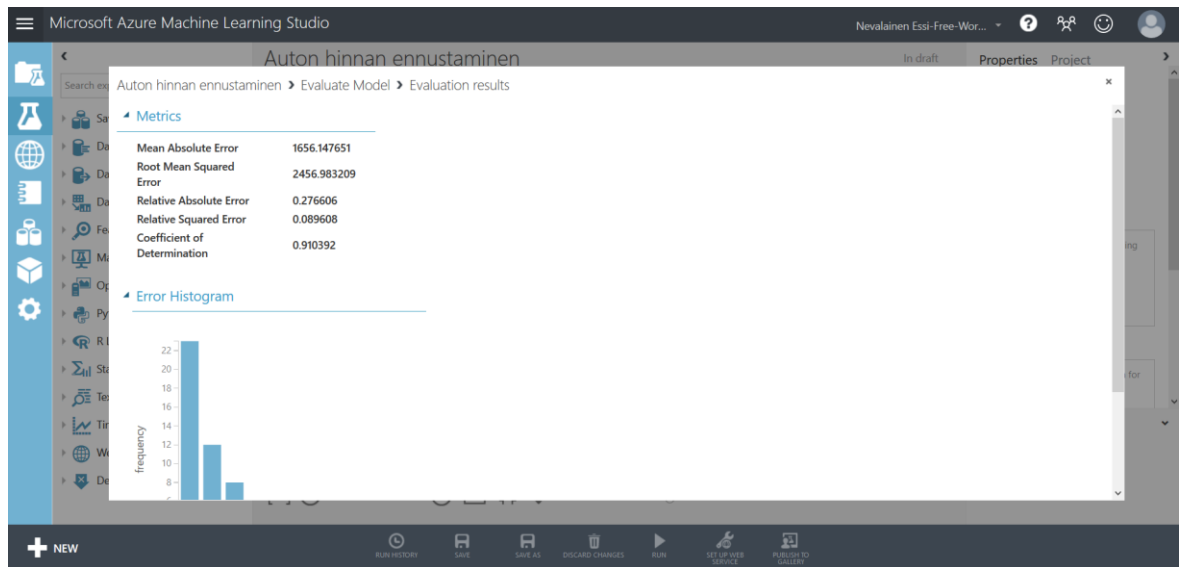
Kuva 3. Dataa käsitellään ja ennusteelle olennaiset ominaisuudet valitaan

Tässä vaiheessa data jaetaan mallin opetus- ja testausdataan. Opettamisessa käytetään 75 % datasta ja loput 25 % varataan testaamisvaiheeseen. Lisäksi valitaan oppimisalgoritmi. Tässä prosessiketjussa se on lineaarinen regressio -moduuli. Regressio ennustaa numeerista arvoa, ja lineaarinen regressio pyrkii luomaan lineaarisen suhteen riippumattomien muuttujien ja tulosmuuttujan tai riippuvaisen muuttujan välillä, joka on myös numeerinen (Microsoft 2017i). (Kuva 4.) (Microsoft 2017h.)

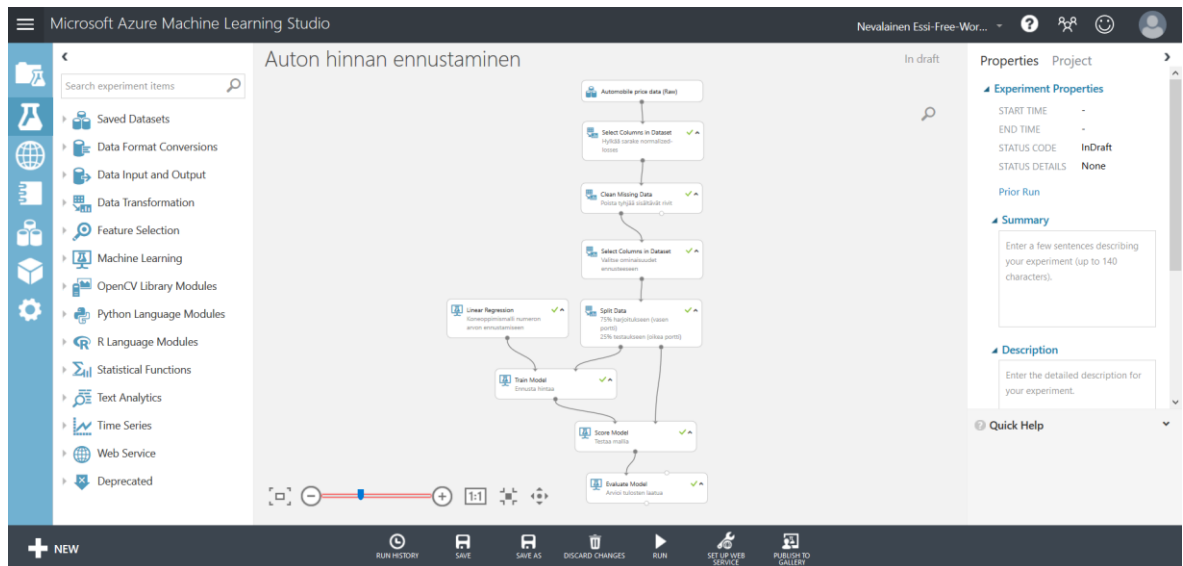


Kuva 4. Prosessiketjussa edetään datan jakamiseen ja valitaan oppimisalgoritmi

Kun malli on opetettu ennustamaan auton hintaa käyttämällä 75 % datasta, testataan mallia 25 %-illa. Lopuksi arvioidaan tulosten laatua. Visualisointi-ikkunassa näytetään mallin tilastot, joista voidaan päätellä kuinka hyvin ennuste toimii (ks. kuva 5). Mitä lähempänä esimerkiksi determinatiokerroin on arvoa 1, sitä parempi ennuste on. Kuvassa 6 on näkymä valmiista prosessiketjusta. (Microsoft 2017h.)



Kuva 5. Tulosten arvioiminen tilastojen perusteella



Kuva 6. Näkymä valmiista prosessiketjusta Microsoft Azure Machine Learning Studiassa

5 Ohjelmistorobotin työtehtävät

Opinnäytetyön toiminnalliseen osaan kuului ohjelmistorobotin työtehtävien kuvaaminen. Jotta tehtäviä voitaisiin sijoittaa kalenteriin ja kalenteria pystyttäisiin optimoimaan, oli alkuun määriteltävä mitä tietoja tehtävään voi liittyä.

Määrittely alkoi selvittämällä yhteistyössä toimeksiantajan kanssa, millaisia vaatimuksia työtehtävään voi resursoinnin näkökulmasta liittyä. Näitä ovat esimerkiksi työtehtävän kesto, prioriteetti ja toistuvuus. Ajankohtaan liittyvä tarve aikatauluttaa tehtävät sujuvasti ja poikkeustilanteet huomioiden.

Määrittelyn tuloksena syntyi taulukko ohjelmistorobotin työtehtävien parametreista (ks. taulukko 2 alaluvussa 5.1 Ohjelmistorobotin työtehtävien parametrit). Taulukon parametreja voidaan käyttää tehtävän tietoina, kun tehtäviä sijoitellaan kalenteriin.

Määrittelyvaiheessa tähtäimenä oli tehtävien automaattisen optimoimisen mahdollistaminen. Sen vuoksi mukaan otettiin myös optimointia parantavia parametreja jotka eivät olleet tehtävän kuvaamiselle pakollisia. Toisaalta taulukko haluttiin pitää tiiviinä ja karsia pois ylimääräiset sekä epäolennaiset parametrit.

Esimerkiksi ohjelmistorobotin työtehtävien tilanhallinnalla ei ole kalenteriin sijoittelussa tai optimoinnissa suurta merkitystä, sillä olennaisiksi voidaan ajatella kolme tilaa: odottaa suoritusta, suorituksessa ja suoritettu. Tämän vuoksi tila jätettiin pois parametreista.

Myös tehtävän kuvaus jätettiin määrittelyn aikana pois taulukosta, vaikka sille tunnistettiin jonkin tason arvo työntekijälle. Jos olisi esimerkiksi tarve ymmärtää sijoittelematta jääneiden tehtävien listaa lukemalla nopeasti mistä tehtävistä olikaan kyse, voisi kuvaava teksti prosessista ja kohdejärjestelmistä auttaa. Tässä vaiheessa päädyttiin kuitenkin karsimaan kuvaus-parametri pois taulukosta.

Pois jätettiin myös tehtävän kohde sekä työhön tarvittavat resurssit ja erityisvaatimukset. Nämä asiat määritellään ohjelmistorobottien prosessimäärityksen dokumentaatioissa, eikä niitä tarvita tehtävän aikatauluttamisessa kalenteriin. Jatkokehityksessä asiaa voidaan tarkastella uudelleen.

5.1 Ohjelmistorobotin työtehtävien parametrit

Valituista parametreista koostettiin taulukko. Ohjelmistoroboteille annettaviin työtehtäviin liittyvät parametrit jaettiin kahteen kategoriaan: pakollisiin ja lisäarvoa tuottaviin. Parametreja avataan tarkemmin omissa alaluvuissaan.

Taulukko 2. Ohjelmistorobotin työtehtävien parametrit

Parametri	Kuvaus	Pakollinen	Tuottaa lisäarvoa
Tehtävän ID	Yksilöivä tunniste	x	
Tehtävän nimi	Selkokieline nimi	x	
Kuormitustaso	Kuinka monta prosenttia kapasiteettia tehtävä vaatii ohjelmistorobotilta		x
Toistuvuus	Miten usein tehtävä toistetaan	x	
Kesto	Arvio tehtävän suorittamiseen kuluva ajasta	x	
Aloituspvm	Tehtävä alkaa tänä päivämääränä	x	
Aloitusklo	Tehtävä alkaa tähän kellonaikaan	x	
Päättymispvm	Tehtävä päättyy tänä päivämääränä		x
Päättymisklo	Tehtävä päättyy tähän kellonaikaan		x
Alkaa aikaisintaan pvm	Tehtävä voidaan aloittaa aikaisintaan tänä päivämääränä		x
Alkaa aikaisintaan klo	Tehtävä voidaan aloittaa aikaisintaan tähän kellonaikaan		x
Alkaa viimeistään pvm	Tehtävä tulee aloittaa viimeistään tänä päivämääränä		x
Alkaa viimeistään klo	Tehtävä tulee aloittaa viimeistään tähän kellonaikaan		x
Päättyy aikaisintaan pvm	Tehtävä voidaan päättää aikaisintaan tänä päivämääränä		x
Päättyy aikaisintaan klo	Tehtävä voidaan päättää aikaisintaan tähän kellonaikaan		x
Päättyy viimeistään pvm	Tehtävä tulee päättää viimeistään tänä päivämääränä		x
Päättyy viimeistään klo	Tehtävä tulee päättää viimeistään tähän kellonaikaan		x
Prioriteetti	Tehtävälle annettu tärkeysaste		x
Robotin ID	Määrittelee mille ohjelmistorobotiyksikölle tehtävä tulee osoittaa		x
Varaus	Ohjelmistorobottikapasiteetti varataan aikaleimojen väliselle ajalle. Varaus kertoo, kuinka monta työtehtävää (1–n) varaudutaan tekemään ajan puitteissa. Mikäli arvo on muu kuin n, voidaan varaus purkaa, kun suoritettavien tehtävien lukumäärä on täyttynyt. Yksittäisen työtehtävän kesto saadaan kesto-parametrilla.		x

5.2 Pakolliset parametrit

Pakollisia parametreja tarvitaan aina, kun työtä halutaan sijoittaa kalenteriin. Näitä parametreja tunnistettiin määrittelyssä kuusi kappaletta, joita avataan tässä luvussa seuraavaksi tarkemmin.

Tehtävällä tulee olla **ID**, jonka avulla se on yksilöitävissä. Tehtävälle on lisäksi annettava selkokielen **nimi**, esimerkiksi ”Ostolaskujen täsmäytys”, jonka avulla ihmisen on tarvittaessa helpompi osallistua työvuorosuunnitteluun.

Toistuvuus-parametri mahdollistaa kertaalleen määritellyn tehtävän toistamisen halutulla tavalla. Toistuvia tehtäviä ovat esimerkiksi henkilöstö- ja taloushallintoon liittyvät raportit, sekä säännöllisesti toistuviin prosesseihin kuten palkanmaksuun liittyvät tarkistukset ja täsmäytykset. Joitakin tehtäviä toistetaan päivittäin, kun taas osa saattaa liittyä esimerkiksi viikon tai kuukauden vaihtumiseen.

Tehtävän **kesto** on arvio, jota mukautetaan tarvittaessa. Jos tiedossa on esimerkiksi merkittävä kasvu käsiteltävien rivien volyymissa, on kestoa hyvä sopeuttaa ennakoituun tilanteeseen. Pakollisista parametreista kaksi viimeisintä määrittävät tehtävän **aloittamisen ajankohdan**. Tehtävälle annetaan päivämäärä ja kellonaika, jonka koittaessa ohjelmistorobotti aloittaa tehtävän suorittamisen.

Nämä äsken kuvatut kuusi pakollista parametria ovat ihmiselle vielä helposti hallittavia. Vastaavia tehtäviin liittyviä määritteitä onkin käytössä useissa kalenterisovelluksissa, esimerkiksi Microsoftin Outlook-kalenterissa, kun käyttäjä luo uuden tehtävän tai tapaamisen.

Pelkkien pakollisten parametrien avulla työtehtävien optimointi jää kuitenkin melko alkeelliselle tasolle, sillä niitä käytettäessä kaikki tehtävät ovat samanarvoisia ja asetettu alkaamaan juuri tietynä ajankohtana. Tämänkaltaiset tehtävät eivät mahdollista sitä joustoa tehtävien ajoittamisessa, mitä työtehtävien sijoittelutyökalu vaatisi toimiakseen optimaalisesti. Tämän vuoksi määrittelyvaiheessa tunnistettiin myös lisäarvoa tuottavia parametreja.

5.3 Lisäarvoa tuottavat parametrit

Lisäarvoa tuottavat parametrit parantavat työtehtävien optimaalista sijoittelua kalenteriin. Ne eivät ole pakollisia tehtävää luotaessa. Näiden parametrien tehokas käyttö on ihmiselle haastavaa ja hidasta useiden erilaisten muuttujien vuoksi, sillä huomioon on otettava yhtä aikaa kaikkien listalla olevien työtehtävien keskenään riitelevät parametrit.

Algoritmipohjaisessa käsittelyssä sen sijaan lukuisten erilaisten rajoitteiden samanaikainen huomioiminen on mahdollista ja nopeaa. Lisäarvoa tuottavia parametreja tunnistettiin määrittelyssä neljätoista kappaletta, joita avataan tässä luvussa seuraavaksi tarkemmin.

Kuormitustaso-parametrin avulla määritellään, kuinka monta prosenttia kapasiteettia tehtävä vaatii ohjelmistorobotilta. Osa ohjelmistorobottituotteista perustuu robottiyksiköiden sijaan virtuaalipalvelinkapasiteetin käyttöön. Tällaisten ohjelmistorobottien kohdalla on olennaista arvioida virtuaalipalvelimen kuormitusaste per tehtävä, koska tehtäviä voidaan ajaa rinnakkain.

Päätymisen päivämäärä ja kellonaika ovat valinnaisia parametreja. Määriteltäessä toettiin aloitusajan olevan ihmiselle päättymisaikaa luontevampi käyttää, jonka vuoksi aloitusaika asetettiin pakolliseksi. Päätymiseen liittyviä parametreja voidaan käyttää esimerkiksi varausta tehdessä. Varauksesta kerrotaan tarkemmin lisää myöhemmin tässä luvussa.

Toinen esimerkki päätymisajan parametrien käytöstä on jouston luominen tehtävän loppupäähän (ks. kuva 7). Asettamalla päättymisaika arvioitua kestoä myöhäisemmäksi, on ohjelmistorobotille varattu lisäääää siltä varalta, että tehtävän suorituksessa kestääkin arvioitua pidempään. Syynä voi olla esimerkiksi järjestelmien hitaus tai vastaamattomuus.



Kuva 7. Päätymisajan parametrien avulla tehtävän loppuun voidaan luoda lisääää

Muita **aloittamiseen ja päätymiseen** liittyviä lisäarvoa tuottavia parametreja tunnistettiin useita. Tehtävällä voi olla joustoa optimointiin tuovia arvoja *aikaisintaan* ja *viimeistään* sen aloitus- ja päättymisajoille. Näiden parametrien avulla mahdollistetaan yhtä lukkoon lyötyä aloitus- tai päättymisaikaa suurempi liikkumavara tehtävien sijoittamisessa. Jokainen aikaparametri jakaantuu vielä erikseen päivämäärään ja kellonaikaan.

Esimerkiksi kuvassa 8 olevan tehtävän aloitusta voidaan siirtää yhden tunnin sisällä tarpeen mukaan. Näin lähekkäin osuvien tehtävien aikataulutusta on mahdollista hienosäätää aiempaa paremmin keskenään.



Kuva 8. Lisäarvoa tuottavien parametrien aloitus- ja päättymisajat tuovat joustavuutta tehtävien aikataulutukseen

Prioriteetti-parametrin avulla voidaan kuvata, onko tehtävä liiketoiminnan kannalta kriittinen. Lisäarvoa tämä parametri tuottaa mahdollistamalla tehtävien painottamisen keskenään. Jos samasta kalenteripaikasta kilpailee kaksi eri tehtävää, voidaan alemmalla prioriteettiasteella oleva sijoittaa muualle.

Robotin ID -parametrin avulla voidaan tarvittaessa määritellä mille ohjelmistorobottiyksikölle tehtävä tulee osoittaa. Syitä voivat olla esimerkiksi robottiyksikön osaaminen tai tarve lainata lisäkapasiteettia toiselta yksiköltä. Parametri identifioi robottiyksiköt, koska useissa ohjelmistorobottiikkatuotteissa yksikkö kykenee käsittelemään vain yhtä tehtävää kerrallaan. Tästä rajoitteesta johtuen robottiyksiköillä on yksilölliset työjonot. Identifioimalla ohjelmistorobottiyksiköt, voidaan niille kuitenkin rakentaa yhteinen varauskalenteri tai muu työjonojen hallintatyökalu.

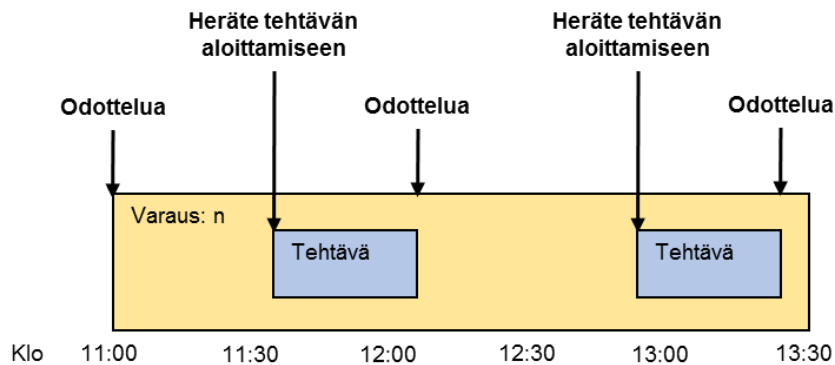
Lisäksi jos yksittäinen tehtävä pitää sisällään n kappaletta tietorivejä, sama tehtävä voidaan jakaa useille robottiyksiköille siten, että rivit jaetaan useiden yksiköiden hoidettavaksi. Esimerkiksi Blue Prism -tuotteen työjonojen hallinta mahdollistaa tehtävän jakamisen vaikkapa kolmelle eri ohjelmistorobotille.

Varaus on viimeinen lisäarvoa tuottava parametri. Ohjelmistorobotin kapasiteetti varataan tehtävän alkamisen ja päättymisen väliselle ajalle. Varaus-parametri kertoo, kuinka monta työtehtävää (1– n) varaudutaan tekemään varausajan puitteissa.

Jos varaus-parametrin arvo on muu kuin n , voidaan varaus purkaa, kun suoritettavien tehtävien lukumäärä on täyttynyt. Tämä vapauttaa ohjelmistorobotin kapasiteetin muuhun käyttöön. Jos varaus-parametrin arvo taas on esimerkiksi 3, ja kaikki kolme tehtävää ovat

suoritettuina jo varauksen alkupuolella, voi ohjelmistorobotti ottaa jäljellä olevalle ajalle muita tehtäviä.

Jos arvo taas on n , ohjelmistorobotti odottaa koko varausajan purkamatta varausta ja tekee tämän ajan sille annettuja tehtäviä. Kuvassa 9 havainnollistetaan kalenterivarausta, jonka aikana ohjelmistorobotti on varautunut tekemään n kappaletta tehtäviä. Odottelun katkaisevat herätteet tehtävien aloittamiseen.



Kuva 9. Varaus-parametri arvolla n luo kalenterivarauksen, jonka aikana varaudutaan tekemään n kappaletta tehtäviä

Varausta voidaan käyttää esimerkiksi ihmisen tai järjestelmän käynnistämisen tehtävän ajamiseen, jolloin heräte voi olla manuaalinen tai automaattinen. Syitä varauksen käytölle voivat olla esimerkiksi tarve odottaa jonkin toisen tehtävän tai eräajon valmistumista.

Varausta voidaan käyttää myös asiakaspalvelun apuna. Kun asiakas soittaa puhelintukeen, voi asiakaspalvelija käynnistää ohjelmistorobotin louhimaan datamassasta vastauksen ongelmaan keskustelun aikana (Lacity & Willcocks 2016b, 48–49). Varauksen aikana suoritettavan yksittäisen työtehtävän kesto saadaan kesto-parametrilla.

5.4 Työtehtävien optimointi

Kun ohjelmistorobotin työtehtäville on olemassa malli, jolla ne voi kuvata ja sijoittaa kalenteriin, voidaan kalenteria ryhtyä tiivistämään tehokkaammaksi. Opinnäytetyössä lähtökohdana oli ajatus koneoppimisen hyödyntämisestä kalenterin automaattisessa optimoinnissa.

Työtehtävien optimoinnille olennaiseksi tunnistettiin tilastotiedot ohjelmistorobottien töiden toteutuneista työajoista. Ohjelmistorobotiikkatuotteet keräävät monenlaista dataa automaattisesti, mikä mahdollistaa tilastotiedon keräämisen koneoppimisen käyttöä varten.

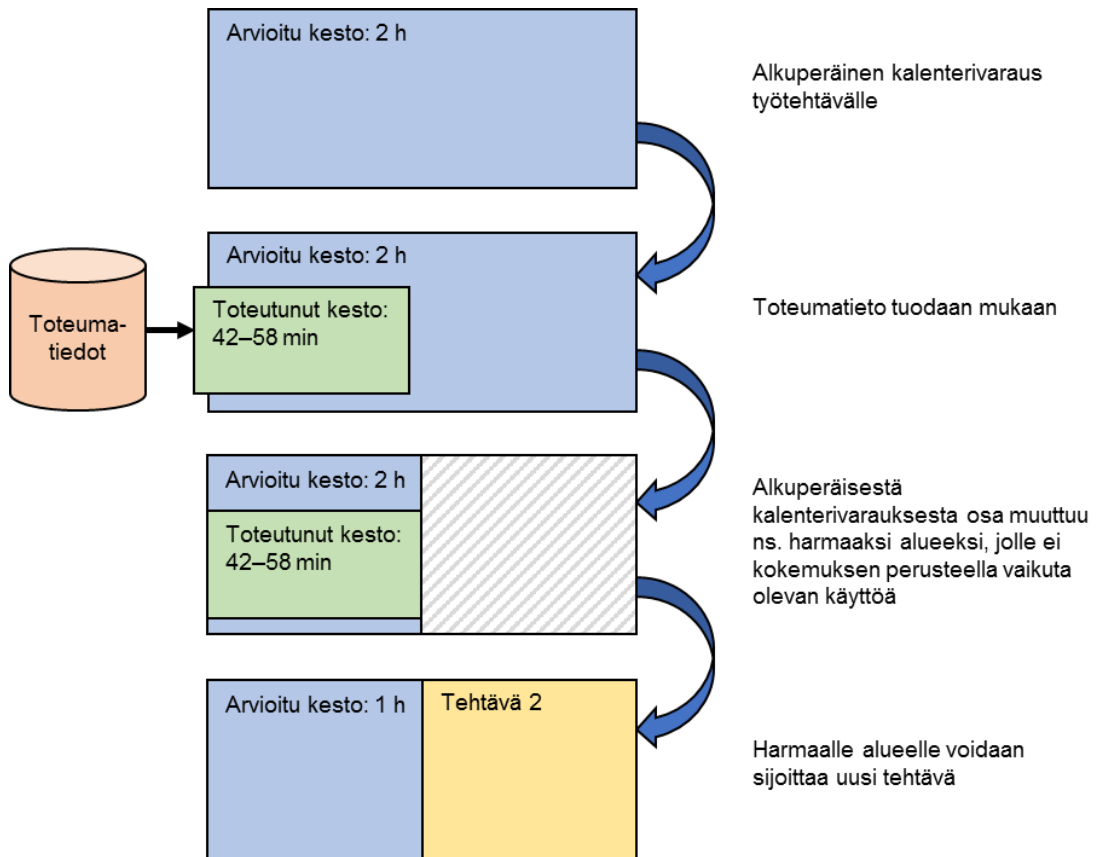
Esimerkiksi Blue Prism mahdollistaa jokaisen ohjelmistorobotin tekemän asian tallentamisen lokeihin. Näin laaja tietojen kerääminen ei ole kuitenkaan järkevää. Tiedot vievät paljon tilaa, minkä lisäksi tietoturvaan ja tietosuojaan liittyvät asiat rajoittavat tiedon keräämistä. Blue Prismissä tavalliset tilastotiedot tulevat vakiona työjonon tietoihin. Niistä on nähtävissä muun muassa keskimääräisiä ja yksittäisiä aikoja, sekä tehtävän aloitus- ja valmistumisaikoja. (Apostolakis & Takkula 5.4.2017.)

Tilasto- eli toteumatiedon pohjalta koneoppiminen pystyy havaitsemaan, milloin tehtävän suoritus vie aikaa enemmän tai vähemmän kuin sille on alun perin määritetty. Tämän tiedon avulla tehtävä voidaan määrittellä keston osalta uudelleen. Toteumatiedosta pystytään havaitsemaan myös se, että arvioitu kesto on jo linjassa toteutuneen kanssa.

Koneoppimisen avulla pystytään ennustamaan niin ikään tilanteita, joissa esimerkiksi joka toinen päivä suoritettava raportin ajaminen kestää kuukauden ensimmäisellä viikolla 38 % normaalia pidempään. Kun tämä on selvinnyt, suuremman rasituksen ajalle osuvat tehtävät voidaan sopeuttaa todelliseen kestoonsa.

Ennustavan koneoppimisen avulla tehtäviä voidaan sijoittaa kalenteriin hyvinkin tiiviisti. Jos ihminen on arvioinut työtehtävän kestoksi kaksi tuntia mutta ohjelmistorobotti on toistuvasti kuluttanut siihen aikaa todellisuudessa vain alle tunnin, voidaan ennustaa asian olevan näin jatkossakin. Tämä mahdollistaa tehtävien sijoittelun päällekkäin tai niiden keston lyhentämisen toteuman mukaiseen ajanjaksoon, ylimitoitettun aika-arvion sijaan.

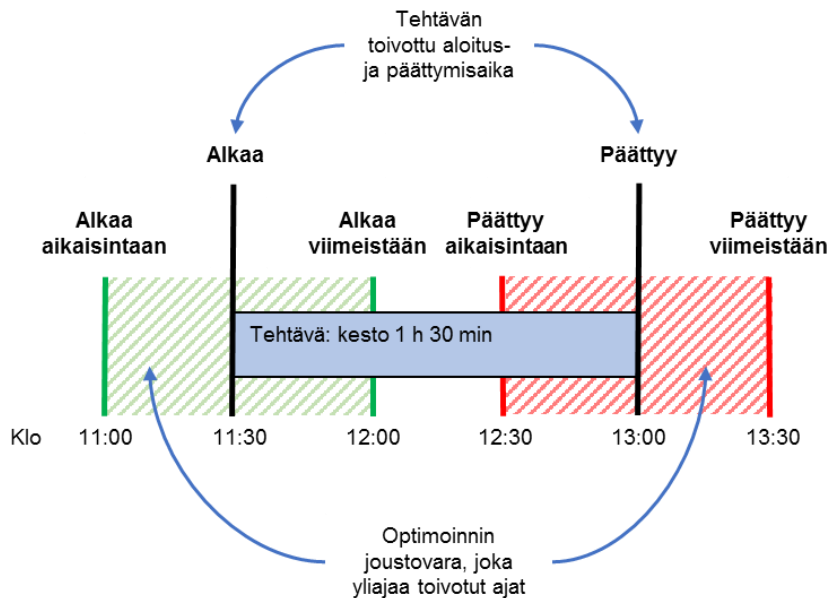
Toteumatieto synnyttää kalenteriin sijoitettujen työtehtävien välille ns. harmaita alueita (ks. kuva 10). Tällä tarkoitetaan aikaikkunoita, jotka eivät ensi silmäyksellä vaikuta olevan hyödynnettävissä. Kalenterissa ei esimerkiksi ole riittävästi tilaa uusille tehtäville, koska ne menisivät keskenään osittain päällekkäin. Tällaisten harmaiden alueiden tehokas hyödyntäminen edellyttää toteutuneiden työtehtävien tilastollista analysointia.



Kuva 10. Toteumatietojen hyödyntäminen ohjelmistorobotin työkalenterin tiivistämisessä

Jotta lisäarvoa tuottavia parametreja voidaan käyttää rinnakkain pakollisten parametrien kanssa, on niiden välinen suhde määriteltävä työtehtävien sijoittelutyökalulle. Jos tehtävällä on aloitus- ja päättymisaikaan liittyviä arvoja *aikaisintaan* ja *viimeistään*, yliajavat ne seuraavat joustamattomat parametrit: aloitus pvm, aloitus klo, päätyminen pvm ja päättyminen klo.

Näitä joustamattomia aloitus- ja päättymisaikoja voidaan ajatella tehtävän toivottuna suoritusajana. Optimoinnin mahdollistamiseksi on kuitenkin määriteltävä, että joustavat parametrit ohittavat toivotun suoritusajan. Jos optimoinnin vuoksi on tarvetta siirtää tehtävää toivotusta suoritusajasta, se voidaan tehdä optimoinnin joustovaran sisällä (ks. kuva 11).



Kuva 11. Eri aikaparametrien painottaminen keskenään

Aikaparametrien lisäksi optimointia parantavat robotin ID ja tehtävän prioriteetti. Robotin ID -parametrin avulla voidaan optimoida ohjelmistorobottien kokonaiskäyttöä hyödyntämällä tasaisesti eri robottiyksiköitä.

Prioriteetti-parametri taas toimii tehtävien välisenä arvottajana, jossa korkeamman prioriteetin tehtävä voittaa kilpailun kalenteripaikasta. Jos kaksi saman prioriteetin tehtävää kuitenkin kilpailee samasta kalenteripaikasta, syntyy konflikti. Tilanteessa jossa muut robottiyksiköt eivät voi ottaa toista tehtävää, on sijoittelutyökalun nostettava kummatkin listalle, josta ihminen voi sijoittaa ne manuaalisesti kalenteriin.

Lisäarvoa tuottavien parametrien määrittely on lisätyö tehtävän määrittelyssä, joka saattaa aiheuttaa muutoksia organisaation toimintamalleihin, aikatauluksiin ja prosesseihin. Liiketoiminnan asiantuntijan on luotava säännöt ennen kenties vain hiljaisena tietona olevista asioista, kuten tehtävän prioriteetista tai aikaisimmasta mahdollisesta aloitus- ja lopetusajankohdasta. Prosessien automatisoiminen tarjoaakin organisaatiolle tilaisuuden miettiä uusia toimintatapoja.

5.5 Parametrien jatkokehitys ja ylläpito

Ohjelmistorobotin työtehtävien parametrit -taulukkoa voidaan kehittää jatkossa vielä tarkemmalle tasolle. Tarkkuuden etuna voi olla entistä tehokkaampi optimointi ja parempi käytettävyys. Jatkokehitystä tapahtuu luultavasti jo sijoittelutyökalua tehtäessä, jolloin havaitaan mahdolliset muutostarpeet. Kehitystä tapahtunee myös käytännön asiakastyön

myötä, jolloin paranee ymmärrys siitä, mikä on olennaista ja mitä parametreista tällä hetkellä saattaa puuttua.

Yksi mahdollinen kehityskohde on toistuvuus-parametri. Koska erilaisia toistuvuuskuvioita on lukuisia, on sitä varten kenties kehitettävä oma laajempi määrittelyosionsa. Toistumisessa tulisi huomioida myös arkipyhät ja muut poikkeukset, esimerkiksi järjestelmien huoltokatkot. Vaikka pyhäpäivät eivät vaikutakaan ohjelmistorobottiin, voivat ne vaikuttaa järjestelmien käytettävyyteen ja sitä kautta myös ohjelmistorobotin työhön (Apostolakis & Takkula 5.4.2017).

Myös kuvaus-parametri voi nousta myöhemmin tarpeelliseksi, jolloin se kannattaa lisätä taulukkoon. Kuvaustekstin avulla työntekijän olisi luultavasti helpompaa ymmärtää ja muistaa mistä tehtävässä oli kyse. Tämä nousee luultavasti enemmän esiin, kun tehtäviä on asiakkaalla enemmän kuin kymmenittäin. Jatkokehityksessä voidaan pohtia myös sitä, ovatko erityisvaatimukset ja tarvittavat resurssit olennaisia määrittellä parametreina.

Yksi kehitysehdotus on myös verkkolomakkeen luominen parametrien perusteella, johon asiakas voisi kirjata tehtävän tiedot suoraan. Lomakkeen etuna voivat olla parempi käyttöösi ja asiakaskokemus. Lomakkeesta tiedot voisivat siirtyä oikeassa muodossa koneoppimisympäristöön tai tietokantaan.

Lopputuloksen ylläpitotarve on vähäinen. Taulukkoa on pidettävä ajan tasalla, mikäli parametreihin tulee muutoksia. Taulukko ja sitä avaavat tekstit toimivat tukena sijoittelutyökalun kehittämiseksi, ohjelmistorobotiikan kohteiden määrittelijälle ja ohjelmistorobotin toteuttajalle.

6 Ohjelmistorobottikapasiteetin käyttö- ja varauskalenterin malli

Ohjelmistorobotin työtehtävien parametrien määrittelyn jälkeen seuraava vaihe oli suunnitella käyttö- ja varauskalenteri, johon tehtäviä sijoitettaisiin. Kalenterin tarkoitus on visualisoida ohjelmistoroboteille suunniteltua työkuormaa ja näyttää selkeästi, missä kohdassa on tilaa sekä mihin aikaan kapasiteettia käytetään enemmän.

Työtehtävien sijoittelutyökalu ei tarvitsisi kalenteria avukseen, mutta koska ihmiset suunnittelevat ohjelmistorobotin käyttöajat, on visuaalinen näkymä luetteloa helpompi käsitellä. Kalenteria voidaan käyttää myös toteutuneen käytön esittämiseen asiakkaalle.

6.1 Tehtävien vienti kalenteriin tehtäväluettelon avulla

Aiemmin kuvattujen parametrien avulla luodaan ohjelmistorobottien työtehtäviä, jotka taas kootaan tehtäväluetteloksi. Tiedot voivat tulla luetteloon esimerkiksi verkkolomakkeelta, tietokannasta tai manuaalisesti lisättynä.

Luettelossa olevat tehtävät voidaan viedä kalenteriin täysin manuaalisesti, osittain manuaalisesti tai automaattisesti. Jos ihminen syöttää tehtävät kalenteriin manuaalisesti, ei lisäarvoa tuottavia parametreja kuitenkaan kyetä hyödyntämään täysimääräisesti. Lisäksi työ on hidasta, virheeltistä ja vie työaikaa tuottavammalta tekemiseltä.

Tehtävien sijoitleminen kalenteriin kannattaakin automatisoida ohjelmistorobotiikan tai koneoppimisen avulla. Luettelon rivien vieminen kalenteriin voidaan opettaa ohjelmistorobotille yhtenä tehtävänä. Jos lisäarvoa tuottavia parametreja halutaan hyödyntää, tehtävien sijoitleminen kalenteriin voidaan suorittaa koneoppimisympäristössä. Koneoppimisympäristössä voidaankin ennustamisen lisäksi tehdä myös muuta, kuten asettaa työtehtäviä kalenteriin.

Opinnäytetyön toiminnallisessa osassa suunniteltiin esimerkki tehtäväluettelosta, jolla tehtäviä voidaan kirjata parametrien avulla aikataulutettavaksi (ks. liite 1). Kaikki luettelon kohdat eivät ole työtehtävien sijoittelutyökalulle olennaisia, mutta ne auttavat ihmistä tunnistamaan tehtävän. Luettelon sarakkeet ovat ohjelmistorobotin työtehtävien parametreja vastaavia.

Tehtäväluetteloon lisättiin tyyppi-sarake, jota ei ole listattu parametreissa. Kun sarakkeessa lukee työ, ohjelmistorobotti lähtee heti käyntiin. Jos sarakkeessa lukee varaus, ohjelmistorobotti jää odottamaan herätettä tehtävän aloittamiseen. Varausarvo-sarake ilmaisee

suoritettavien tehtävien lukumäärän varausaikana. Liitteessä 1 on myös testidataa sisältävä esimerkki tehtäväluettelosta.

6.2 Kalenterimalli

Käyttö- ja varauskalenteri tehtiin toimeksiantajan toiveesta Exceliin. Kyseessä on yksi esimerkki siitä, miten visuaalisen kalenterin voi toteuttaa (ks. liite 2 ja 3). Kalenterista toteutettiin kaksi versiota: toisessa tehtäväkeston minimiyksiköksi asetettiin 15 minuuttia ja toisessa 30 minuuttia.

Kun tehtävien sijoittelussa käytetään automaatiota, voi kalenterissa käytettävä ajan minimiyksikkö olla näitä pienempikin. Esimerkiksi minuutin pituinen aikayksikkö sallisi tarkemman aikataulutuksen. Projektissa päädyttiin kuitenkin aloittamaan yksiköillä, jotka olivat ihmiselle paremmin hallittavia.

Kuvassa 12 näkyy kuvakaappaus kalenterista, jossa on näkyvissä yksi aikataulutettu työtehtävä. Kalenterin sarake A kertoo kuukauden päivän ja sarake B viikonpäivän. Rivillä 1 ovat vuorokauden kellonajat, kuvan esimerkissä 30 minuutin yksikköinä. Aikataulutettu tehtävä on merkitty tunnuksella 1_T1, jossa 1 tarkoittaa solun statusta. Statuksia on kolme: 0 on vapaa, 1 on työ ja 2 on varaus. Tehtävän tunnuksen loppuosa T1 on tehtävän ID.

	A	B	C	D	E	F
1			00:00	00:30	01:00	01:30
2	1	to	0	0	0	0
3	2	pe	1_T5	1_T5	1_T5	0
4	3	la	0	0	0	0
5	4	su	0	0	0	0
6	5	ma	0	0	0	0
7	6	ti	0	0	0	0

Kuva 12. Esimerkki kalenterista testisisällöllä

Solujen taustaväreillä voidaan myös ilmaista solun statusta, jolloin ihmisen on helpompi tulkita kalenteria. Vihreä ilmaisee vapaata, keltainen varaus-tyyppistä tehtävää ja punainen työ-tyyppistä tehtävää. Jotta tiettyinä viikonpäivinä toistuvat tehtävät olisivat helpompia aikatauluttaa kalenteriin, sarakkeessa B ovat viikonpäivät.

6.3 Kalenterin ja tehtäväluettelon jatkokehitys sekä ylläpito

Kalenteri ja tehtäväluettelo tulevat todennäköisesti muuttumaan, kun työtehtävien sijoitteluyökalua ryhdytään kehittämään. Kyseessä onkin alustava malli, jonka pohjalta voidaan aloittaa kehittäminen ja ajatusten testaaminen.

Yksi kehitysehdotus on siirtyminen pienempiin aikayksikköihin kalenterissa. Kun tarkoituksena on se, ettei ihminen täytä manuaalisesti kalenteria, on 15 minuuttia aikayksikkönä turhan pitkä koneelliselle kalenterin hyödyntämiselle. Jatkokehityksessä kalenteri voidaan myös toteuttaa tietojärjestelmän sisälle, jolloin siitä voidaan vain tarvittaessa generoida Exceleitä ihmiskäyttöä varten.

Lopputulosten ylläpitotarve on vähäinen. Jos tehtäväluetteloon tai kalenteriin tulee muutoksia, on versioita päivitettävä tarvittavilta osin.

7 Pohdinta

Tässä luvussa käydään läpi opinnäytetyöprojektin lopputulos, tehdään johtopäätökset ja esitellään jatkokehitystehtävät. Luvussa perustellaan myös valittuja menetelmiä, käydään projektikonaisuus läpi sekä arvioidaan onnistumista. Luvun lopussa pohditaan tulosten ajankohtaisuutta ja tarpeellisuutta sekä mietitään oppimiskokemusta ja ammatillista kehittymistä.

7.1 Lopputuloksen arvioiminen ja johtopäätökset

Projektin konkreettisia tuotoksia ovat taulukko ohjelmistorobotin työtehtävien parametreista, tehtäväluettelo sekä varaus- ja käyttökalenteri. Muita tuloksia ovat kirjallinen osuus, johon kuuluvat näitä tuotoksia tarkemmin avaavat osuudet ja yleinen tietoperusta.

Vastauksena johdannossa esitettyyn kysymykseen, ”voidaanko ennustavaa koneoppimista hyödyntää ohjelmistorobotiikan resurssien optimoinnissa”, voidaan vastata kyllä. Koneoppimista voidaan hyödyntää optimoinnissa. Ensimmäinen ja oleellisin perustelu on koneoppimisen kyky tehdä ennusteita toteumadatan perusteella. Näitä ennusteita voidaan käyttää optimoinnissa.

Toinen perustelu on se, että koneoppimisella saadaan joustava ja lähes rajaton laskentakapasiteetti, jonka avulla kyetään käsittelemään suuriakin tietomassoja. Kolmas perustelu koneoppimisen hyödyntämiselle on se, että koneoppimispalvelut ovat helposti julkaistavissa www-sovelluspalvelumuodossa.

Tarkempi näkemys asiaan saadaan vasta kun lopputuloksia hyödynnetään käytännön asiakastyössä, mutta alustavasti tulos on lupaava. Lopputulos ei ole yllättävä, sillä se on vahvistus ennako-olettamalle.

Opinnäytetyön tehtävänasetteluna oli myös selvittää, miten ohjelmistorobottiresurssien optimointia voidaan automatisoida. Tähän vastattiin miettimällä, miten ohjelmistorobottien tehtäviä voidaan kirjata ylös ja sijoitella kalenteriin manuaalisesti. Koska koneoppiminen tarjoaa automatisoidun itsestään kehittyvän keinon käsitellä hetkessä suuria datamassoja, todettiin yhdeksi sen käyttösovellukseksi ohjelmistorobotin työkalenterin täyttäminen.

Projektin lopputulosta voidaan mielestäni pitää luotettavana. Tietoperustan muodostamiseen käytettyjä lähteitä koostaessani pyrin kiinnittämään huomiota niiden monipuolisuuteen ja tieteellisyyteen. Hankalaksi osoittautui kuitenkin löytää etenkin ohjelmistorobotii-

kasta riippumattomia ja tieteellisiä lähteitä kovinkaan monipuolisesti. Markkinointisävyteistä materiaalia oli kyllä saatavilla, mutta koska kyseessä on verrattain tuore teknologia, ei siitä ole tehty paljoakaan tieteellistä tutkimusta. Lisäksi alan nopea muutostahti asetti oman haasteensa lähteiden hankintaan. Asiantuntijoille tehdyt haastattelut antoivatkin arvokkaan lisän tietoperustaan.

Opinnäytetyön johtopäätös on siis se, että ohjelmistorobottikapasiteettia kannattaa kokeilla optimoida koneoppimisen avulla. Suosittelen aloittamaan kehittämisen kevyesti ja tekemään sen iteroiden, jolloin saadaan nopeasti tuloksia ja tunnistettuihin puutteisiin sekä kehitystarpeisiin voidaan reagoida sukkelasti.

7.2 Jatkokehitystehtävät

Opinnäytetyön aikana kehitetyt ohjelmistorobotin työtehtävien parametrit, tehtäväluettelo sekä käyttö- ja varauskalenteri ovat malleja, joiden pohjalta voidaan aloittaa varsinainen työkalun kehittäminen ja ajatusten testaaminen. Vasta käytännön kehitystyön aikana havaitaan todelliset puutteet ja jatkokehitystarpeet. Muutamia kehitysideoita nousi projektin aikana kuitenkin esiin.

Ohjelmistorobotin työtehtävien parametrit -taulukkoa voidaan kehittää jatkossa nyt määritettyä tarkemmalle tasolle. Tarkkuuden etuna voi olla entistä tehokkaampi optimointi ja parempi käytettävyys. Yksi mahdollinen kehityskohde on toistuvuus-parametri. Koska erilaisia toistuvuuskuvioita on lukuisia, on toistuvuutta varten mahdollisesti kehitettävä oma laajempi määrittelyosionsa.

Taulukkoon voi olla tarpeellista lisätä myös kuvaus-parametri. Kuvaustekstin avulla ihmisen olisi helpompaa ymmärtää ja muistaa mistä tehtävässä oli kyse. Jatkokehityksessä voidaan myös miettiä sitä, onko erityisvaatimuksille luotava oma parametri, jonka avulla ilmaistaisiin esimerkiksi tehtävien välisiä riippuvuuksia tai tiedossa olevia erikoistilanteita.

Yksi kehitysehdotuksista on myös verkkolomakkeen luominen parametrien perusteella. Lomakkeen etuna voivat olla Exceliä parempi käyttöliittymä ja asiakaskokemus. Lomakkeesta tiedot voisivat siirtyä oikeassa muodossa suoraan koneoppimisympäristöön tai tietokantaan.

Kalenteria koskeviin kehitysehdotuksiin kuuluu siirtyminen pienempiin aikayksikköihin varauskalenterissa. Nykyiset 15 ja 30 minuuttia ovat turhan pitkiä koneelliselle kalenterin hyödyntämiselle. Niillä on kuitenkin hyvä aloittaa niiden selkeyden vuoksi, koska manuaa-

lista käyttöä tulee olemaan alussa enemmän. Toinen idea on kalenterin toteuttaminen tietojärjestelmän sisälle, jolloin siitä voidaan vain tarvittaessa generoida Exceleitä ihmis-käyttöä varten.

7.3 Projektin kulku ja erityispiirteet

Projekti alkoi tietoperustan koostamisella kirjallisista lähteistä. Se toimi hyvänä pohjustuksena projektin tuotosten suunnitteluvaiheeseen, jota tehtiin rinnakkain tietoperustan jatkamisen kanssa. Projektin loppuvaiheessa jatkettiin tietoperustan ja tuotoksia avaavien osuuksien kanssa.

Projektin menetelmät toimivat mielestäni tarkoituksenmukaisesti. Puolistrukturoiduilla asiantuntijahaastatteluilta saatiin lisänäkökulmia tietoperustaan. Haastattelulaji valittiin, koska teemahaastattelu olisi ollut tarkoitukseen liian lavea, eikä strukturoidulla haastattelulla oltaisi saatu riittävästi keskustelua aiheesta ja sen vierestä. Toisen haastattelun toteuttaminen parihaastatteluna toimi hyvin.

Ohjelmistorobotin työtehtävien parametrien määrittelyssä toimittiin ideoiden työpajamaisesti. Vertailuanalyysia tehtiin myös kalenterisovelluksiin. Tehtäväluetteloa ja kalenteria tehtäessä ei haettu esimerkkiä kirjallisuudesta.

Toimeksiantajayhteistyö MOST Digital Oy:n kanssa toimi hyvin läpi projektin. Toimeksiantaja tarjosi aktiivisesti asiantuntemustaan ja osallistui lopputulosten suunnitteluun. Työelämävetoinen opinnäytetyöprojekti oli motivoiva, sillä tekemiselle oli alusta alkaen asiakasnäkökulma ja tuloksilla oli jatkokehitykselle annettavaa. MOST Digital Oy on yksi uranuurtaja Suomessa opinnäytetyön toimialoilla, ja projektin aikana oli havainnollistavaa nähdä myös yrityksen näkökulmasta alan nopea muutostahti.

7.4 Tulosten ajankohtaisuus ja tarpeellisuus

Opinnäytetyön käsittelemät aiheet ovat tekemässä läpimurtoa Suomessa, ja ala kehittyy nopeaa vauhtia (Tivi 2017). Ohjelmistorobotiikka ja koneoppiminen olivatkin hyvin ajankohtaisia aihevalintoja opinnäytetyölle.

Opinnäytetyö pohjustaa työkalun kehitystä työtehtävien automaattiseen aikatauluttamiseen ja optimointiin. Tällaisella työkalulla voi olla merkittävää vaikutusta ohjelmistorobottien hyödyntämiseen matalalla kustannustasolla, jolla taas on vaikutusta projektien ja palveluiden hinnoitteluun. Edullinen ohjelmistorobotiikka voi lisätä teknologian käyttöä eri

liiketoiminta-alueilla, tehostaa prosesseja sekä mahdollistaa ihmisille tuottavamman sekä mielekkäämmän työn tekemisen.

Asiantuntijahaastatteluissa nousi esiin se, että automaattiselle työtehtävien aikataulutukseen ja optimoinnille on kasvava tarve tulevaisuudessa. Mitä enemmän työtä automatisoidaan ja robotisoidaan, sitä suuremmaksi optimointitarve nousee kustannusten kasvun myötä. Vaikka työkalu olisi pelkästään automaattisesti sijoitteleva eikä optimoiva, tuottaisi se luultavasti tarpeeksi ajallista säästöä suunnitteluun, jotta sellaisen toteuttaminen olisi kannattavaa.

Opinnäytetyön tuloksia voidaan hyödyntää jo nyt toimeksiantajan asiakastyössä ja jatkokehittämisessä. Projekti tarjoaa toimeksiantajalle vahvistuksen ennako-olettamaan siitä, että ohjelmistorobottikapasiteettia voidaan optimoida koneoppimisen avulla. Opinnäytetyön tuloksia voivat hyödyntää myös ohjelmistorobotiikan kohteiden määrittelijät ja ohjelmistorobotin toteuttajat työssään.

7.5 Oppiminen ja ammatillinen kehittyminen

Opinnäytetyön aikana projektinhallinnallinen osaamiseni kehittyi etenkin laajan kokonaisuuden osittamisen ja aikataulutuksen osalta. Saavutin myös paremman ymmärryksen ohjelmistorobotiikasta ja koneoppimisesta. Projektin aikana perehdyin niihin teknologioihin ja tuotteisiin, sekä syvennyin kummankin käyttömahdollisuuksiin opinnäytetyön tehtävänasettelun näkökulmasta.

Projektia jälkikäteen tarkastellessa yhtenä kehityskohtana esiin nousee projektin kokonaisaikataulutus. Alkuperäinen aikataulu oli suunniteltu liian lyhyeksi tekemiseen nähden. Kun tämä havaittiin, projektin suuntaa ja aikataulua muutettiin onnistuneesti. Lisäksi kun kokemusta on nyt kertynyt arvioidusta ajankäytöstä projektivaiheittain ja tehtävittäin, on vastaisuudessa helpompi arvioida osuvammin tuntien allokointi projektisuunnitelmaan.

Kokonaisuutena projekti oli mielenkiintoinen ja ammatillisesti hyvin kehittävä. Uskon saaneeni opinnäytetyöprosessista osaamista, jota voin jatkossa kehittää edelleen työelämässä. Ohjelmistorobotiikka ja koneoppiminen ovat jo nyt täällä, eivätkä ne ole hetkeen katoamassa – kaikesta oppimastani tulee olemaan minulle hyötyä työelämässä.

Lähteet

Agrawal, A., Gans, J. & Goldfarb, A. 2017. What to Expect From Artificial Intelligence. MIT Sloan Management Review, 58, 3, s. 23–27.

Apostolakis, T. & Takkula, A. 5.4.2017. Ratkaisukonsultti. Digital Workforce Services Oy. Haastattelu. Helsinki.

Arntz, M., Gregory, T. & Zierahn, U. 2016. The Risk of Automation for Jobs in OECD Countries: A Comparative Analysis. OECD Social, Employment and Migration Working Papers, 189. OECD Publishing. Paris. Luettavissa: <http://dx.doi.org/10.1787/5jlz9h56dvq7-en>. Luettu: 15.2.2017.

Blue Prism 2016. Blue Prism confirms its position as the world's leading provider of Robotic Process Automation (RPA) software, quadrupling the rate of new business with 96 new customers. Luettavissa: <https://www.blueprism.com/7305>. Luettu: 10.2.2017.

Chappell, D. 2015. Introducing Azure Machine Learning – A Guide for Technical Professionals. Luettavissa: http://www.davidchappell.com/writing/white_papers/Introducing-Azure-ML-v1.0--Chappell.pdf. Luettu: 21.2.2017.

Davenport, T. & Kirby, J. 2016. Just How Smart Are Smart Machines? MIT Sloan Management Review, 57, 3, s. 21–25.

Digital Illustrated 2015. Mitä Azure Machine Learning ja koneoppiminen mahdollistavat käytännössä? Luettavissa: <http://www.digitalillustrated.com/Ajankohtaista/Mita-Azure-Machine-Learning-ja-koneoppiminen-mahdollistavat-kaytannossa/>. Luettu: 21.4.2017.

Espoon kaupunki 2017. Kotihoidon Ropsu-robotti palkittiin kansainvälisessä innovaatiokilpailussa. Luettavissa: [http://www.espoo.fi/fi-FI/Sosiaali_ja_terveyspalvelut/Kotihoidon_Ropsurobotti_palkittiin_kansa\(113487\)](http://www.espoo.fi/fi-FI/Sosiaali_ja_terveyspalvelut/Kotihoidon_Ropsurobotti_palkittiin_kansa(113487)). Luettu: 17.4.2017.

Festum 2017. Ohjelmistorobotiikka – RPA. Luettavissa: <http://www.festum.fi/fi/software/tuotteet/kofax/kofax-kapow.html>. Luettu: 17.4.2017.

Frey, C. & Osborne, M. 2013. The Future of Employment: How susceptible are jobs to computerisation? University of Oxford. Oxford. Luettavissa:

http://www.oxfordmartin.ox.ac.uk/downloads/academic/The_Future_of_Employment.pdf.

Luettu: 15.2.2017.

Gartner 2016. Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage. Luettavissa:

<http://www.gartner.com/newsroom/id/3412017>. Luettu: 13.2.2017.

InfoWorld 2016. How IBM, Google, Microsoft, and Amazon do machine learning in the cloud. Luettavissa: <http://www.infoworld.com/article/3039052/cloud-computing/how-ibm-google-microsoft-and-amazon-do-machine-learning-in-the-cloud.html>. Luettu: 20.4.2017.

Karlos, A., Kujala, J & Martinsuo, M. 2006. Projektiliiketoiminta. WSOY Oppimateriaalit Oy. Helsinki.

Lacity, M. & Willcocks, L. 2016a. Service Automation: Robots and the Future of Work. Steve Brookes Publishing. Stratford-upon-Avon.

Lacity, M. & Willcocks, L. 2016b. A New Approach to Automating Services. MIT Sloan Management Review, 58, 1, s. 41–49.

Louhia 2016. Azure ML – Prediktiivisen analytiikan pilvipalveluiden aatelinen? (vertailu IBM Watson analyticiin). Luettavissa: <http://www.louhia.fi/2016/05/16/azure-ml-prediktiivisen-analytiikan-pilvipalveluiden-aatelas-vertailu-ibm-watson-analyticiin/>. Luettu: 21.4.2017.

Lyytinen, J. 2017. Tekoälytutkija Timo Honkela sairastui parantumattomasti – nyt hän haluaa kertoa meille tulevaisuudesta, jota ei itse tule näkemään. Helsingin Sanomat, 34, s. B2–B5.

Microsoft 2017a. Introduction to Azure Machine Learning in the cloud. Luettavissa: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-machine-learning>. Luettu: 19.4.2017.

Microsoft 2017b. Machine learning algorithm cheat sheet for Microsoft Azure Machine Learning Studio. Luettavissa: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-cheat-sheet>. Luettu: 19.4.2017.

Microsoft 2017c. Azure Machine Learning frequently asked questions: Billing, capabilities, limitations, and support. Luettavissa: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-faq>. Luettu: 20.4.2017.

Microsoft 2017d. Execute Python machine learning scripts in Azure Machine Learning Studio. Luettavissa: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-execute-python-scripts>. Luettu: 21.4.2017.

Microsoft 2017e. Author custom R modules in Azure Machine Learning. Luettavissa: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-custom-r-modules>. Luettu: 21.4.2017.

Microsoft 2017f. Products available by region. Luettavissa: <https://azure.microsoft.com/en-us/regions/services/>. Luettu: 20.4.2017.

Microsoft 2017g. Microsoft tukee organisaatioiden valmistautumista EU:n tietosuojasetukseen. Luettavissa: <https://news.microsoft.com/fi-fi/2017/02/15/microsoft-tukee-organisaatioiden-valmistautumista-eun-tietosuojaasetukseen/#sm.000008vp53snefee5rt1f5k2lcs42>. Luettu: 20.4.2017.

Microsoft 2017h. Machine learning tutorial: Create your first data science experiment in Azure Machine Learning Studio. Luettavissa: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-create-experiment>. Luettu: 21.4.2017.

Microsoft 2017i. Linear Regression. Luettavissa: <https://msdn.microsoft.com/library/azure/31960a6f-789b-4cf7-88d6-2e1152c0bd1a/>. Luettu: 21.4.2017.

MOST Digital Oy 2016. Luettavissa: <http://www.mostdigital.fi>. Luettu: 17.4.2017.

Munoz, A. 2014. Machine Learning and Optimization. Luettavissa: https://www.cims.nyu.edu/~munoz/files/ml_optimization.pdf. Luettu: 19.4.2017.

Niikko, A. 2015. Kuntien kiinteistönhoidon resurssien mitoituksen optimointi. Pro gradu - tutkielma. Aalto-yliopisto. Insinööritieteiden korkeakoulu. Luettavissa: <http://urn.fi/URN:NBN:fi:aalto-201512165645>. Luettu: 23.4.2017.

Pajarinen, M. & Rouvinen, P. 2014. Computerization Threatens One Third of Finnish Employment. ETLA. 22. Luettavissa: <https://www.etla.fi/wp-content/uploads/ETLA-Muistio-Brief-22.pdf>. Luettu: 15.2.2017.

Palkeet 2016. Palkeet näyttää suuntaa ohjelmistorobotiikassa. Luettavissa: <http://www.palkeet.fi/ajankohtaista/palkeet-nayttaa-suuntaa-ohjelmistorobotiikassa.html>. Luettu: 17.4.2017.

Paukku, T. 2016. Algoritmi tuntee tapasi. Helsingin Sanomat, 66, s. B8–B9.

Pelin, R. 2011. Projektihallinnan käsikirja. Projektijohtaminen Oy Risto Pelin. Helsinki.

Saarinen, J. 2016. Kehittävätkö tietokoneet tietoisuuden? Totta kai, sanoo suomalainen tekoäly-yrittäjä. Helsingin Sanomat, 98, s. A24–A25.

SikuliX 2017. Luettavissa: <http://sikulix.com>. Luettu: 17.4.2017.

Säisä, S. 21.4.2017. Osakas ja konsultti. MOST Digital Oy. Haastattelu. Espoo.

Tietotekniikan termitalkoot 2012. Www-sovelluspalvelu. Luettavissa: <http://www.tsk.fi/tsk/termitalkoot/fi/node/266>. Luettu: 19.4.2017.

Tivi 2017. Ohjelmistorobotiikka on kuuma ala: "Suomalaisilla on Euroopassa hyvät asetelmat". Luettavissa: http://www.tivi.fi/Kaikki_uutiset/ohjelmistorobotiikka-on-kuuma-ala-suomalaisilla-on-euroopassa-hyvät-asetelmat-6620246. Luettu: 16.4.2017.

UiPath 2017. Strategic Partners. Luettavissa: <https://www.uipath.com/partners>. Luettu: 13.2.2017.

Liitteet

Liite 1. Tehtäväluettelon malli tyhjänä ja esimerkkisisällöllä

Tehtävän ID	Tehtävän nimi	Kuormitusaste	Toistuvuus	Kesto	Prioriteetti	Tyyppi	Varausavo	Robotti ID	Alkaa PVM alkaistaan	Alkaa KLO alkaistaan	Päättyy PVM alkaistaan	Alkaa KLO viimeistään	Päättyy PVM viimeistään	Päättyy KLO alkaistaan	Päättyy KLO viimeistään	Toivottu aloitus PVM	Toivottu aloitus KLO	Toivottu päättymisen PVM	Toivottu päättymisen KLO	

Tehtävän ID	Tehtävän nimi	Kuormitusaste	Toistuvuus	Kesto	Prioriteetti	Tyyppi	Varausavo	Robotti ID	Alkaa PVM alkaistaan	Alkaa KLO alkaistaan	Päättyy PVM alkaistaan	Alkaa KLO viimeistään	Päättyy PVM viimeistään	Päättyy KLO alkaistaan	Päättyy KLO viimeistään	Toivottu aloitus PVM	Toivottu aloitus KLO	Toivottu päättymisen PVM	Toivottu päättymisen KLO
T1	Käyttäjätunnusten luominen	100	Päivittäin	00:30	3 Matala	Työ		Mika tahansa								10.6.2017	12:00	10.6.2017	14:00
T2	Raporttiero	25	Päivittäin	00:30	1 Korkea	Varaus	n	Robotti A								1.6.2017	08:00	1.6.2017	12:00
T3	Tilauksen tekemisen	50	Keskiviikko	01:30	2 Normaalii	Työ		Robotti B	7.6.2017	01:00		04:00							
T4	Lomalistaus	100	Sunnuntai	03:00	1 Korkea	Työ		Robotti B	4.6.2017	16:30		22:00				4.6.2017	16:30		
T5	Täsmäytys	100	Ei toistoa	04:30	2 Normaalii	Työ		Mika tahansa	1.6.2017	20:00	2.6.2017	23:00	2.6.2017	00:30	03:30	1.6.2017	21:00		

Liite 2. Kalenterimalli tyhjänä

	00:00	00:15	00:30	00:45	01:00	01:15	01:30	01:45	02:00	02:15	02:30	02:45	03:00	03:15	03:30	03:45	04:00
1 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4 su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5 ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6 ti	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7 ke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11 su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12 ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13 ti	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14 ke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18 su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19 ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20 ti	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21 ke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25 su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26 ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27 ti	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28 ke	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Liite 3. Kalenterimalli esimerkkisisällöllä

	00:00	00:30	01:00	01:30	02:00	02:30	03:00	03:30	04:00	04:30	05:00	05:30	06:00	06:30	07:00	07:30	08:00	08:30	09:00	09:30	10:00	10:30	11:00	11:30	12:00
1 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
2 pe	1_T5	1_T5	1_T5	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
3 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
4 su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
5 ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
6 ti	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
7 ke	0	0	0	0	1_T3	1_T3	1_T3	1_T3	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
8 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
9 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
10 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
11 su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
12 ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
13 ti	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
14 ke	0	0	0	0	1_T3	1_T3	1_T3	1_T3	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	1_T1
15 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
16 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
17 la	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
18 su	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
19 ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
20 ti	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
21 ke	0	0	0	0	1_T3	1_T3	1_T3	1_T3	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
22 to	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0
23 pe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	2_T2	0