



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Jesse Vauhkonen

Lähetysseuran tietohallinnon prosessien automatisointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tutkinto-ohjelma

Insinöörityö

6.5.2020

Tekijä Otsikko	Jesse Vauhkonen Lähetysseuran tietohallinnon prosessien automatisointi
Sivumäärä Aika	24 sivua 6.5.2020
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintäteknikka
Ammatillinen pääaine	Ohjelmistotuotanto
Ohjaajat	Yliopettaja Auvo Häkkinen Järjestelmäasiantuntija Henri Siponen
<p>Insinööriyön tarkoituksena oli tutustua ohjelmistorobotiikkaan, sen työkaluun UiPathiin, ja toteuttaa sen avulla rutiininomaisen prosessin automatisointi. Työ tehtiin Suomen Lähetysseuralle, joka työllistää yli 200 henkilöä eri puolilla maailmaa.</p> <p>Lähetysseuran tietohallinto on jo pitkään pyrkinyt automatisoimaan yksinkertaisia työtehtäviään. Viime aikoina hyväksi todettu ohjelmistorobotiikan työkalu UiPath on auttanut automatisoimaan myös vaativampia kokonaisuuksia. Insinööriyön tavoitteena oli vähentää rutiininomaista manuaalista työtä tietokoneille tehtävien asennusjonojen luomisessa. Työssä myös selvitettiin, kuinka monimutkaisia prosesseja voidaan automatisoida ennen kuin siihen vaaditaan ihminen osaksi.</p> <p>Työssä perehdyttiin Lähetysseuran tietohallinnon prosesseihin ja nykyiseen olemassa olevaan automaatioon sekä hyvän automaation käytäntöihin. Lopussa suunniteltiin ja toteutettiin automaatio tietokoneiden asennusjonojen luomiseen ja huomattiin, että se säästää noin 20-30 työtuntia vuodessa. Tulevaisuudessa se voi säästää enemmän, koska kynnys tehdä asennusjonoja on matalampi.</p>	
Avainsanat	ohjelmistorobotiikka, UiPath, automatisointi

Author Title	Jesse Vauhkonen Automating FELM's IT-Processes
Number of Pages Date	24 pages 6 May 2020
Degree	Bachelor of Engineering
Degree Programme	Information- and Communications Technology
Professional Major	Software Development
Instructors	Auvo Häkkinen, Senior teacher Henri Siponen, Systems specialist
<p>The purpose of this bachelor's thesis was to get acquainted with robotic process automation, its tool UiPath, and to implement one automation for a routine process. The work was done for the Finnish Evangelical Lutheran Mission, FELM, which employs more than 200 people around the world.</p> <p>FELM's information management has been automating its simple tasks for some time now. The recently found robotics tool UiPath has helped in automating some routine processes already. The aim of this bachelor's study was to reduce routine manual work in creating installation queues for computers. This study also explored how complex processes can be automated before a human part is required.</p> <p>The work focused on FELM's IT-processes and existing automation, as well as good automation practices. In the end, automation was created and implemented to create computer installation queues and it was found that it saves about 20-30 man-hours per year. In the future, it may save more because the threshold for making installation queues is lower.</p>	
Keywords	Robotic Process Automation, UiPath, automation

Sisällys

Lyhenteet

1	Johdanto	1
2	Ohjelmistorobotiikka	2
2.1	Ohjelmistorobotti	2
2.2	Hyödyntäminen	3
2.3	Virheiden hallinta ja raportointi	5
3	UiPath-sovellus	8
3.1	Toiminta	8
3.2	Vertailu kilpailijoihin	10
4	Toteutuksen suunnittelu	11
4.1	Microsoft Deployment Toolkit	11
4.2	Asennusjonon rakentaminen	12
4.3	Toiminnallisuuden osiin jakaminen UiPathissa	13
5	Toteutus	15
5.1	Alkutoimenpiteet ja ajurien työstö	15
5.2	Käyttöönotto työkalun ensimmäinen toiminto ja valintaprofiili	17
5.3	Tehtäväsarjat ja lopputoimenpiteet	20
5.4	Testaus	22
6	Tulokset	22
7	Johtopäätökset	23
	Lähteet	24

Lyhenteet

EXE	Suoritettava tiedostomuoto.
HP	HP Inc., entinen Hewlett-Packard, yhdysvaltalainen suuryritys ja laite- sekä ohjelmistovalmistaja. Lähetysseura käyttää pääosin HP:n tietokoneita.
IDE	Integrated Development Environment. Kehitysympäristö, ohjelma, jossa luodaan ohjelmistoja ja koodia.
IPv4	Yleisimmin käytössä oleva internet-protokolla. Yhdistää internetin laitteet palvelimiin ja sieltä käyttäjiin.
MDT	Microsoft Deployment Toolkit, mahdollistaa Windowsin automaattisen tietokoneen asennusjonon verkon välityksellä.
RPA	Robotic Process Automation – ohjelmistorobotiikka.

1 Johdanto

Insinööriyön tarkoituksena on tutustua ohjelmistorobotiikkaan ja sen työkaluun UiPathiin sekä automatisoida yksi Suomen Lähetysseuran tietohallinnon prosessi. Automaation tavoitteena on vähentää manuaalista rutiininomaista työtä ja selvittää, kuinka paljon työtunteja sillä säästetään. Työssä pohditaan myös hyvän automatiikan käytäntöjä ja mihin robotiikkaa kannattaa soveltaa, sekä analysoidaan toteutettavan automatisaation tuloksia.

Suomen Lähetysseura tekee kehitysyhteistyötä, rauhantyötä ja kirkollista työtä 30 maassa työllistäen yli 200 henkilöä. Näin laajalle levittäytyneelle organisaatiolle on tärkeää, että IT-prosessit ja palvelut pelaavat nopeasti, kunnolla ja ilman häiriöitä. Tästä vastaa Lähetysseuran tietohallinto, joka hoitaa muun muassa palveluiden ylläpitoa, laiteasennuksia ja ongelmatilanteita. Tietohallinnossa on hyvän aikaa sitten havahduttu automaation hyötyihin, ja siellä toimiikin jo yksi Powershellillä ohjelmoitu Skypeissä tekstikomennoilla toimiva botti, joka avustaa laajassa kirjossa erilaisia tehtäviä. Viime aikoina tarkastelussa ollut ohjelmistorobotiikka on avannut kuitenkin aivan uuden kaliiperin mahdollisuuksia työtehtävien ja prosessien automatisointiin. Sopivaksi työkaluksi todetulla UiPathilla onkin jo toteutettu muutamia automatisaatioita tutustumismielessä. Tässä insinööriyössä on tarkoituksena syventyä tarkemmin UiPathiin ja toteuttaa automaatio tietokoneiden asennusjonojen luontiin. Asennusjono mahdollistaa tietokoneen alkuasennuksen halutuilla ohjelmilla ja asetuksilla. Tämä valikoitui automatisoitavaksi prosessiksi, koska asennusjonon luominen käsin on rutiinitehtävä, jonka lisäksi Lähetysseura uusii laitemallistoaan lähitulevaisuudessa. Tämän vuoksi uusien asennusjonojen luomiselle tulee lähiaikoina olemaan tarvetta.

2 Ohjelmistorobotiikka

Ohjelmistorobotiikalla (RPA – Robotic Process Automation) tarkoitetaan teknologiaa, jonka avulla automatisoidaan tietokoneella suoritettavia prosesseja. Automatisoitava prosessi on usein rutiininomainen työtehtävä, kuten tietojen siirtäminen eri järjestelmien välillä, jossa ei tapahdu suuria muutoksia eri suorituskerroilla, mutta myös laajempien kokonaisuuksien automatisointi on yhä suuremmissa määrin mahdollista. Suurin osa ohjelmistoroboteista on ohjelmistorobotiikan työkaluilla toteutettuja vakioituun työnkulkuun perustuvia robotteja. Ne noudattavat klikkaus klikkaukselta ennalta määrättyjä sääntöjä ja toimivat samoissa käyttöliittymissä kuin ihmisetkin. Tällainen robotti ei täten tarvitse ohjelmointirajapintoja tai eriäviä tietojärjestelmiä, sillä se pystyy käyttämään samoja työkaluja kuin ihminenkin.

Ohjelmistorobotiikan synty voidaan ajoittaa jonnekin 2000-luvun alkuun, jolloin kyseinen termi ponnahti pinnalle. Vaikka ohjelmistorobotiikka olikin kehittynyt hiljaksen jo ennen 2000-lukua, vakiintui RPA-termi vasta silloin yleiseen käyttöön kuvaamaan tätä teknologiaa. [1.]

Nyky aikaisten työkalujen ja menetelmien vuoksi ohjelmistorobotiikka on saanut viime aikoina rutkasti huomiota, kun yhä useampi yritys on ottanut käyttöönsä erilaisia robotteja automatisoimaan rutiinitehtäviään. Näiden yritysten joukossa on muun muassa yhdysvaltalainen vähittäiskauppajätti Walmart, saksalainen pankki Deutsche Bank ja yksi maailman suurimpia konsulttiyrityksiä Ernst & Young. Automaation toteutuksen ja käyttöönoton helppous ja suhteellisen pienellä vaivalla aikaansaadut ratkaisut kohottavat työtyytyväisyyttä ja säästävät vuodessa kymmeniä, tai jopa satoja työtunteja. Tulevaisuudessa tekoäly ja koneoppiminen tulevat auttamaan robotteja tehtävissä, jotka vaativat monimutkaisempaa ja itsenäisempää päätöksentekoa ja ongelmatilanteiden ratkaisua. [2.]

2.1 Ohjelmistorobotti

Ohjelmistorobotiikan varsinaista työjuhtaa kutsutaan ohjelmistorobotiksi. Ohjelmistorobotti on työasemalle tai palvelinympäristöön asennettava ohjelmisto, joka suorittaa sille ennestään määrättyjä eli ohjelmoituja tehtäviä. Perinteisen botin, kuten sen Powershellillä ohjelmoidun Skype:ssä toimivan botin ja ohjelmistorobotin erona voidaan yleisesti

pitää sitä, että ohjelmistorobotit toimivat näkyvillä samoissa käyttöliittymissä kuin ihmisetkin. Vastaavasti perinteiset botit toimivat taustalla näkymättömissä, eivätkä ne käytä tavallisia käyttöliittymiä. Ohjelmistorobotti voi toimia itsenäisesti tai ihmisen avustamana, jolloin ihminen valvoo sen toimintaa, mutta robotti suorittaa aikaa vievän osuuden. Robottien toimenkuva voi vaihdella sähköpostien lajittelemisesta monimutkaisten komentosarjojen suorittamiseen tietyinä ajankohtana palvelinympäristössä. Ohjelmistorobotti voidaan myös opettaa vertaamaan tietoja järjestelmien välillä sekä tekemään niiden tietojen perusteella päätöksiä, jotka vaikuttavat robotin toimintaan. [3.]

Ohjelmistorobotin ohjelmoimiseen ei nykyaikaisilla työkaluilla tarvita enää laajaa ohjelmointiosaamista. Ohjelmointi hoituu yleensä vuokaavioilla ja suoraan näyttämällä, mistä robotin tulee klikata. Ohjelmointitausta kuitenkin auttaa esimerkiksi kokonaisuuksien hahmottamisen ja robotin virtaviivaisen ja ongelmattoman suorituksen toteuttamisessa.

Ohjelmistorobotin edut perinteiseen automaatioon verrattuna ovat suuria. Siinä missä perinteinen automaatio vaatii vuosien kokemusta ohjelmoinnista ja sovelluskehityksestä, voidaan ohjelmistorobotti tuottaa vain muutamien viikkojen tai kuukausien perehtymisellä nykyaikaiseen ohjelmistorobotiikkaan. Perinteinen automaatio on myös yleensä raskastekoista, ja sen kehitystyö saattaa paikka paikoin keskeyttää muiden järjestelmien toiminnan, kun taas ohjelmistorobotti on verrattain kevyt eikä vaadi mitään integraatioita tai järjestelmien keskeytyksiä luomisvaiheessa.

Ohjelmistorobotti toimii samoissa käyttöliittymissä kuin ihminen, mutta huomattavasti tarkemmin ja väsymättömästi kellon ympäri jokaisena päivänä vuodessa. Se on myös rutiinista nopeampi eikä kyllästy työtehtäviinsä ja se vapauttaa ihmistyöntekijän tekemään merkityksellisempää ja ajatusta vaativaa työtä rutiinin sijaan. Lisäksi ohjelmistorobotti ei tee virheitä, ellei sitä ole ohjelmoitu väärin. Se suorittaa saman prosessin aina täsmällisesti ja virheettömästi. [4.]

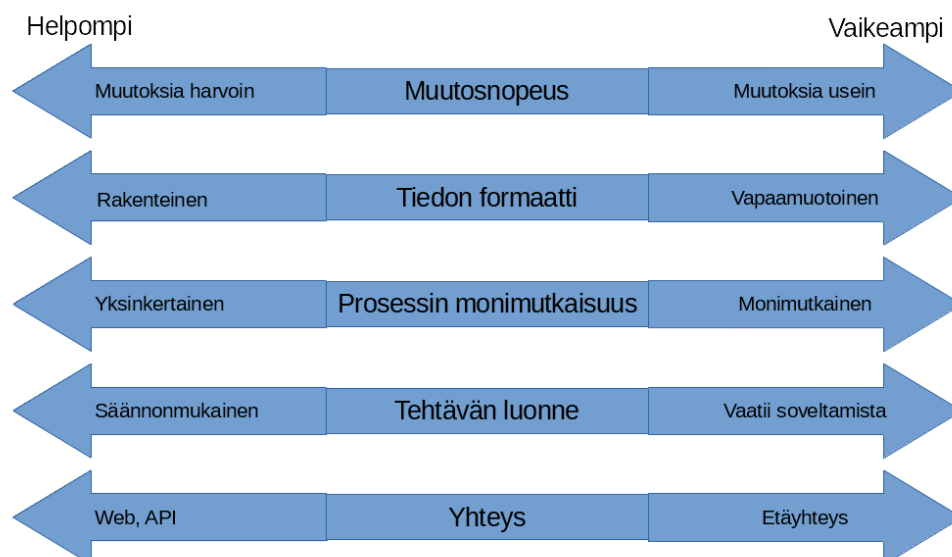
2.2 Hyödyntäminen

Yrityksen tai organisaation kaikkia prosesseja ja työtehtäviä ei tarvitse automatisoida, eikä edes kannatakaan. Ohjelmistorobotiikkaa tulisi enemmänkin hyödyntää yleisesti

kaiken työn teon tehostamiseen rutiinitehtäviä automatisoimalla. Kun automaatiota aletaan miettimään, tulisi myös kiinnittää huomiota muihinkin asioihin, kuten turhien työtehtävien poistamiseen ja poikkeamien minimoimiseen. Ohjelmistorobotiikkaa tulisi käyttää etenkin sen päävahvuuteen: lisätehon tuomiseen rutiineihin.

Automaatiokohteita valittaessa pitäisi kiinnittää huomiota muutamiin seikkoihin, jotka liittyvät oleellisesti robotin toteuttamisen haastavuuteen. Tällaisia asioita ovat työtehtävän muutosnopeus, eli kuinka usein työtehtävän suorituksessa tapahtuu muutoksia, tiedon formaatti, sillä jos se vaihtuu kesken suorituksen, tulisi robotin tunnistaa muutos ja reagoida siihen, prosessin monimutkaisuus, ja tehtävän luonne, eli vaatiiko se soveltamista vai onko se säännönmukainen. Lisäksi tulisi ottaa huomioon, millaisessa ympäristössä robotti toimii, sillä esimerkiksi etäyhteyden kautta toimiva robotti saattaa olla vaikeampi toteuttaa kuin paikallisesti toimiva. On myös huomattava, että ohjelmistorobottien kyky ymmärtää ja hahmottaa esimerkiksi luonnollista kieltä ja kuvia on rajallinen. Robotit ovat parhaimmillaan, kun ne pääsevät valmiiseen, määrämuotoiseen tietoon käsiksi. [5, s. 21.]

Kuvassa 1 on havainnollistettu automaatiokohteiden valinnassa huomioon otettavia asioita helpommin luettavaksi kaavioksi.



Kuva 1. Huomioitavat asiat automaatiokohteiden valinnassa. Keskellä on ominaisuus automatisoitavasta prosessista, ja nuolien päistä voi nähdä, onko ominaisuuden luonne vaikeampi vai helpompi automatisoida. [5, s. 20.]

Kuvassa nuolten keskellä olevat viisi ominaisuutta ovat ne, joihin tulisi kiinnittää huomiota automaatiokohteen valitsemisessa. Nuolten vasemmat puolet kuvaavat ominaisuuden tyyppiä, joka on helpommin automatisoitava, ja oikealla puolella olevat ominaisuuden tyypit ovat vaikeammin toteutettavia. Tämän työn toteutettavan automaation valinta on näillä mittareilla onnistunut. Prosessissa tapahtuu muutoksia harvoin, tiedon formaatti pysyy samana, eikä prosessi ole kovinkaan monimutkainen. Lisäksi tehtävä ei vaadi soveltamista, ja se suoritetaan kokonaan palvelinympäristössä ilman etäyhteyksiä.

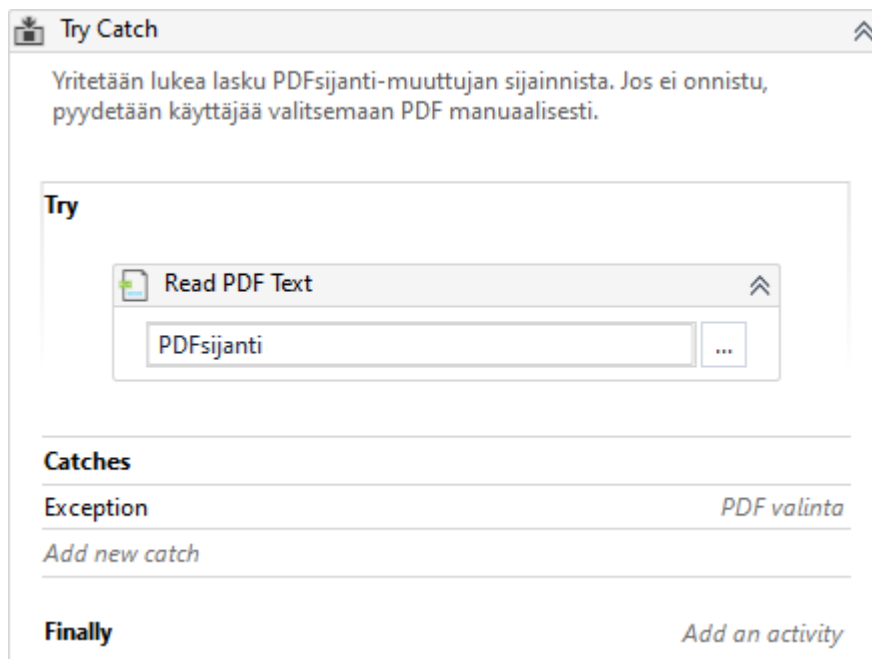
Hyvä käytännön esimerkki automaatiosta on esimerkiksi raportoinnin automatisointi. Lähetysseuran tietohallinnossa toimii UiPathilla toteutettu ohjelmistorobotti, joka kokoaa halutun aikavälin tietoturvaraportin. Robotille annetaan alkuparametreina ajanjakso, esim. tammikuu, ja muut halutut tiedot kuten saapuva roskapostiliikenne, jotka lisätään raporttiin. Sen jälkeen robotti käy läpi tarvittavat järjestelmät ja sivustot sekä kerää kuvankaappauksia Word-dokumenttiin. Loppukäyttäjälle jää lähinnä tietojen tarkastus ja viimeistely. Tällainen automaatio on hyödyllinen, koska se säästää vuodessa yli 20 työtuntia, ja sen kehitykseen kului aikaa noin 3 tuntia. Lisäksi siinä on hyvin pieni mahdollisuus virheisiin, ihmiselle jää kuitenkin viimeinen sana lopputuloksesta.

2.3 Virheiden hallinta ja raportointi

Kuten muissakin ohjelmistoissa ja ohjelmistokehityksessä, myös ohjelmistorobotiikassa tulisi varautua mahdollisiin virheisiin ohjelman suorituksen aikana. Yleensä nämä virhetilanteet jaetaan kahteen kategoriaan: prosessissa tai tehtävässä tapahtuvat muutokset ja tietojärjestelmissä syntyvät vikatilanteet. Ensimmäinen tarkoittaa sitä, että robotin työstämässä prosessissa tai tehtävässä saattaa tapahtua poikkeus, jota robotti ei osaa käsitellä. Tällaisessa tilanteessa varsinaisissa tietojärjestelmissä ei ole mitään vikaa, vaan ongelma on sellainen, joka vaatii poikkeavan tilanteen hahmottamista ja sen perusteella kykyä tehdä päätöksiä, eli toisin sanoen ihmistä. Ideaalitulanteessa robotti on ohjelmoitu merkkamaan ja raportoimaan tällaiset tilanteet, jonka jälkeen se jatkaa omaa suorittamistaan.

Toinen virhekattegoria, tietöjärjestelmissä syntyvät virheet, tarkoittaa virhetilanteita, jotka eivät synny robotin omasta toiminnasta tai prosessista, jota se suorittaa. Esimerkki tällaisesta virheestä on nettisivun latautuminen. Jos robotti käyttää nettisivua joidenkin tietöjen hankkimiseen tai syöttämiseen, mutta nettisivu ei jostain syystä lataudukaan, robotin suoritus keskeytyy. Tällaisiin ongelmiin on hankalampi varautua etukäteen, mutta yksi ratkaisu on laittaa robotti kokeilemaan tietöin väliajoin uudestaan, toimiiko jokin järjestelmä.

Myös koodin puolella on hyvä ottaa huomioon erilaiset häiriötilanteet. Tämä on ohjelmistorobotiikan työkaluilla mahdollista, vaikka ne perustuvatkin vuokaaviokoodaukseen. Esimerkiksi UiPath-ohjelmalla on mahdollista ympäröidä mikä tahansa suoritus perinteisellä Try Catch -menetelmällä, joka ottaa huomioon virheen mahdollisuuden suorituksen aikana ja varautuu siihen vaihtoehtoisella ratkaisulla ilman, että ohjelmisto lopettaa suorituksen kokonaan. Muita hyviä virreehallintatapoja ovat säännöllisten raporttien lähettäminen robotin valvojalle, kattavien lokitiedostöjen pitäminen ja näiden perusteella ajoittaiset päivitykset robotin toimintaan. Kuva 2 näyttää, millainen UiPathin Try Catch -komponentti on. Robotti yrittää lukea tiedostön muuttujan sijainnista, mutta jos se ei onnistu, pyydetään tiedosto valitsemaan manuaalisesti.



Kuva 2. Try Catch -komponentti UiPath-ohjelmassa.

Kuvassa näkyvä toiminto ”Read PDF Text”, joka lukee PDF-tiedoston tekstiä, on ympäröity Try-osiolla. Mikäli tämä osio ei löydä PDF-tiedostoa, siirtyy toiminta Catches-osioon, josta se välitetään edelleen erilliselle PDF-valinta komponentille, jossa käyttäjä valitsee manuaalisesti PDF-tiedoston. Näin vältetään ohjelman kaatumiselta, jos valmis funktio ei löydäkään luettavaa tiedostoa.

Useimmissa työkaluissa on myös sisäänrakennettu järjestelmä, joka lopettaa suorituksen kokonaan virheen sattuessa. UiPathissa suoritus keskeytyy, jos robotti ei 30:een sekuntiin löydä kohdetta, esimerkiksi käyttöliittymän ikkunaa, jota sen tulee käyttää. Tästä lähtee myös raportti ja virheilmoitus käyttäjälle. On kuitenkin mahdollista, että robotti löytää virheellisen tai väärän kentän ja jatkaa suorittamista ohjeiden mukaan. Näiden tapausten varalta on hyvä aina suunnitteluvaiheessa ottaa huomioon kaikenlaiset poikkeukset, jotta jokin pitkiä aikoja yksin toimiva robotti ei tee viikkoja väärää automaatiota. Koska ohjelmistorobotiikka on herkkä muutoksille, on ylipäätään hyvä käytäntö, että kaikista virheistä ja poikkeustilanteista tulisi raportoida robotin ylläpitäjälle tai kehittäjälle, jotta ne voitaisiin jatkokehityksessä noteerata paremmin ja välttää.

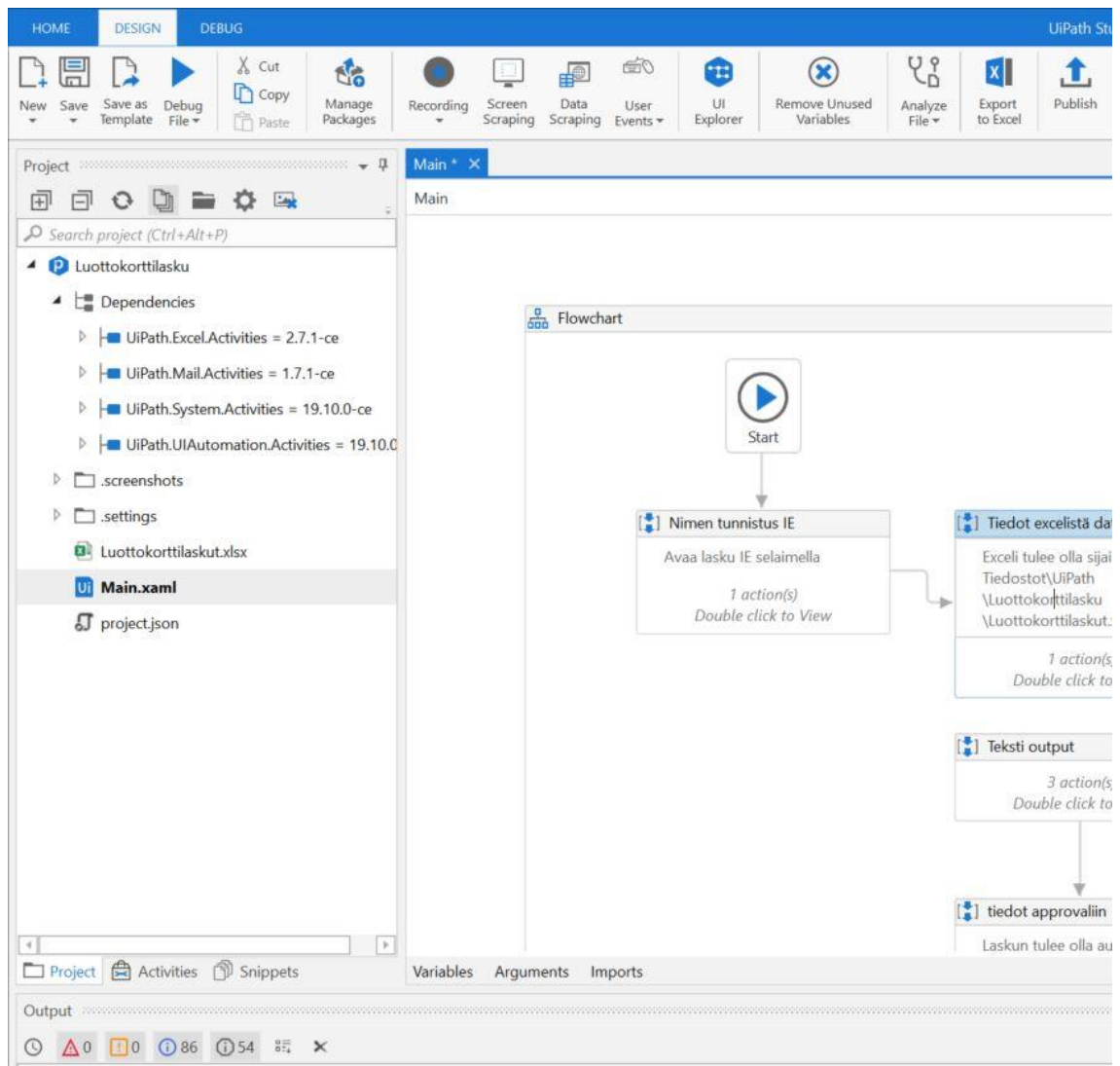
3 UiPath-sovellus

UiPath on vuonna 2005 perustettu ohjelmistoyritys, joka on kehittänyt samaa nimeä kantavan, markkinoiden johtavan ohjelmistorobotiikan työkalun. Tarkkaan ottaen ohjelmiston nimi on UiPath Studio, mutta tässä insinööriyössä siihen viitataan UiPathina. 2010-luvun loppupuolella UiPath on vakaasti kiivennyt ohjelmistorobotiikan suosituimmaksi työkaluksi. Vuoden 2019 viimeisellä neljänneksellä se sai korkeimmat mahdolliset pisteet useissa kategorioissa kaikkien RPA-ohjelmistojen vertailussa. [6.]

UiPathin parhaana puolena pidetään sen käyttäjäystävällisyyttä, sillä sitä voi käyttää tehokkaasti myös ilman ohjelmointitaustaa. Helppokäyttöisyys ei kuitenkaan rajoita sen tehoa: UiPath on yksi markkinoiden tehokkaimpia työkaluja, eikä se hyydy monimutkais-tenkaan prosessien edessä. UiPathista saa ilmaisen Community-version, tai lisäominaisuuksilla varustetun maksullisen Enterprise-version. Enterprise-versiossa on Community-versioon verrattuna muun muassa rajaton määrä suoritettavia robotteja. Tässä insinööriyössä käytössä on Community-versio, joka sopii mainiosti tämän tason automaatioon.

3.1 Toiminta

Toisin kuin tavallisilla koodaustyökaluilla, jossa koodi kirjoitetaan melkein kokonaan käsin, UiPath toimii enimmäkseen vuokaavioilla. Toki mukana on myös tavallisen ohjelmoinnin elementtejä, kuten muuttujia ja luokkia, mutta muuten ohjelma ikään kuin luot itse koodinsa käyttäjän tekemästä suunnitelmasta. Yleiset ominaisuudet ovat kuitenkin melko lailla samat kuin muissakin ohjelmistoympäristöissä, eli IDE:issä. UiPathista löytyvät muun muassa debugging-työkalut, git-ominaisuudet ja virheterminaali. Kokeneemalle ohjelmoijalle UiPath tuntuukin heti melko tutulta ympäristöltä, käytön aloittaminen on varsin helppoa. Eroja tavallisiin ohjelmistoympäristöihin kuitenkin löytyy: UiPathissa on useita ohjelmistorobotiikalle ominaisia työkaluja, kuten toimintojen nauhoittaminen, jossa robotille näytetään, mistä sen tulee klikata, ja datan hakeminen, jossa UiPath hakee esimerkiksi Excel-tiedostosta dataa, jonka se muuttaa itselleen helpommin käytettävään muotoon. Myös toimintojen syvempi muokkaaminen ihan koodin tasolla on mahdollista, jos joitain prosesseja tarvitsee todella yksityiskohtaisesti viilata kuntoon. Kuvassa 3 on osa UiPathin käyttöliittymää.



Kuva 3. UiPathin käyttöliittymää.

Kuvan vasen yläkulma on ohjelman vasen yläkulma. Kuvassa näkyy työkalurivistö ylhäällä, josta löytyy ohjelmistorobotiikalle ominaisia työkaluja, kuten toiminnan nauhoitus sekä käyttöliittymän tutkija. Luokat ja tiedostopuu ovat vasemmalla, ja muutamia robotin toiminnan muodostavia vuokaavioita sijaitsee keskellä ja oikealla. Alhaalla olevaa terminaalialia ei tässä kuvassa näy. Terminaalista voi seurata robotin suorituksen aikaisia tietoja, kuten varoituksia ja virheilmoituksia.

Varsinainen ohjelmistorobotti, joka UiPathilla rakennetaan, toimii UiPath Robot -komponentissa, jonka saa Enterprise-versiolla asennettua tietokoneelle ilman UiPath Stu-

diota. Näin yrityksen koneelle saa robotin operoimaan ilman, että itse ohjelmointiympäristöä tarvitsee asentaa. UiPath Robotissa on mahdollista valita useampi suoritettava automaatio sekä mahdolliset aikataulut, milloin niitä suoritetaan.

3.2 Vertailu kilpailijoihin

UiPath on sijoittunut viime vuosina korkealle useissa RPA-ohjelmistojen vertailuissa. Alan johtavina pidetyt tutkimus- ja konsultointiyhtymät Gartner ja Forrester rankkasivat UiPathin loppuvuodesta 2019 markkinoiden kokonaisvaltaisesti parhaaksi RPA-ohjelmistoksi. [6; 7.] Esimerkiksi Forresterin Forrester Wave -vertailu mittaa kvartaaleittain RPA-ohjelmistojen kykyä vastata asiakkaiden tarpeisiin ohjelmistorobotiikan nopeasti kehittyvässä ympäristössä. Kuvassa 4 on 2019 Q4-kvartaalin eli loppuvuoden Forrester Wave -vertailu.



Kuva 4. Forresterin Q4-2019 vertailu RPA-ohjelmistoista. [6] Vertailu mittaa ohjelmistojen nykyistä kykyä vastata asiakkaiden tarpeisiin sekä tulevaisuuden näkymiä.

Kuvan vaaka-akseli kuvaa yrityksen ja ohjelmiston strategiaa ohjelmiston kehittämisessä ja pystyakseli sitä, kuinka hyvin ohjelmisto pystyy jo nyt vastaamaan alati muuttuviin haasteisiin robotiikan saralla. Ympyrän suuruus kuvaa nykyistä markkinaosuutta. Kuvan ohjelmistot on myös jaettu neljään luokkaan, joita ovat Challengers, Contenders, Strong Performers ja Leaders. Suomennettuna nämä olisivat haastajat, kilpailijat, vahvat suorittajat ja johtajat. Yksi ensimmäisiä ohjelmistorobotiikan työkaluja, Blue Prism, on vieläkin johtavassa asemassa, mutta tarjoaa selvästi heikompaa palvelua kuin UiPath.

UiPathin hyvinä puolina kilpailijoihin verrattuna pidetään sen kykyä yhdistää visuaalinen ja tavanomainen koodaus, eli vuokaaviot ja tavallinen tekstipohjainen koodi. Myös koko ajan suurentuva yhteisö on suuri apu aloitteleville ohjelmoijille, sillä neuvoa saa nopeasti ja helposti. UiPathin yhtenä huonona puolena tai parannettavana asiana on pidetty muuttujien määrittämistä ja säilyttämistä. Muuttujien tyyppejä on rajattu, ja niiden itse määrittäminen on hankalaa, koska on välillä vaikea saada selkoa, minkä tyyppistä muuttujaa UiPath käyttää missäkin tilanteessa.

4 Toteutuksen suunnittelu

Tässä osiossa taustoitetaan ja suunnitellaan toteutettava ohjelmistorobotti. Jotta robotin toimintaperiaatetta voi ymmärtää, on selvitettävä, millaisessa ympäristössä se toimii. Robotti esimerkiksi käyttää Microsoft Deployment Toolkittiä, jolla asennusjonoja luodaan. On hyvä vähän avata sitä ja kertoa, mikä se on ja mitä sillä tehdään. Tämän jälkeen voidaan alkaa suunnittelemaan robotin toimintaa ja toteutustapaa.

4.1 Microsoft Deployment Toolkit

Microsoft Deployment Toolkit, lyhyemmin MDT, on Microsoftin kehittämä työkalu, joka mahdollistaa asennusjonojen luomisen tietokoneiden automaattiseen asennukseen. Asennusjono voi asentaa tietokoneelle valmiiksi halutut ohjelmat, järjestelmänvalvojan tunnuksen, liittää sen haluttuun verkkoon eli domainiin, ja konfiguroida samalla erilaisia asioita, kuten palomuurin asetukset. Se myös asentaa koneille kaikki tarvittavat ajurit. Lähetysseurassa asennusjonot tehdään aina yksittäisen konemallin mukaan. Asennusjonoihin pitää myös aika ajoittain päivittää uudet Windows-asennuspaketit, ettei konetta

tarvitse heti asennuksen jälkeen alkaa päivittää suuresti, jotta uusimmat tietoturvapäivitykset ovat ajan tasalla. Uutta konetta asentaessa asennusjonon valintaan pääsee käynnistyksen lisäasetuksista (Boot Menu) käynnistämällä koneen IPv4-verkkoon, koska Lähetysseuran MDT toimii Windows-palvelimella verkossa.

Deployment Workbench eli käyttöönotto työkalu on yksi MDT:n työkaluista. Sillä hoituu suurin osa tarvittavista toimenpiteistä, joita asennusjonon luomiseksi tarvitsee tehdä. Se vie aikaa manuaalisesti tehtynä ja on rutiininomaista työtä, jossa suurin haaste on lähinnä muistaa kaikki oikeat asetukset ja säännöt, jotka tulee asettaa. Insinööriyön toteutettavaksi automaatioksi valikoituinkin näistä syistä uuden asennusjonon luominen MDT:llä. Tarkoituksena on, että robotille annetaan alussa muutama lähtötieto, kuten haluttu laitemalli, jonka avulla se osaa itse rakentaa asennusjonon halutulle laitteelle. Tämä ohjelmistorobotti ei välttämättä säästä vuodessa kovinkaan paljoa työtunteja, mutta se antaa hyvän pohjan robottien lisäkehitykselle ja pienen selvityksen ohjelmistorobotiikasta ja UiPathista.

4.2 Asennusjonon rakentaminen

Asennusjonon rakentaminen tapahtuu Windows-palvelimella, jossa MDT sijaitsee. Palvelimelle voi ottaa etäyhteyden, mutta UiPathilla on ollut joitain ongelmia toimia etäyhteyden läpi, vaikka se olisikin asennettuna itse palvelimelle. Esimerkiksi jotkin klikkaukset ja näppäinkomennot eivät välttämättä mene läpi. Robotin ohjelmointi suoritetaan siis kokonaan palvelimella, johon onneksi saa näytön, näppäimistön ja hiiren kiinni. Uuden asennusjonon luominen tapahtuu siten, että ensin ladataan laitteeseen sopivat ajurit. Ajurien lataus HP:n laitteilla tapahtuu HP SoftPaq Download Manager -ohjelmalla, tai suoraan HP:n sivustolta. Olisi hyvä myös ladata uusin Windows-versio, mutta tässä toteutuksessa sitä ei tehdä robotilla, koska uusin versio on vastikään ladattu. Sen lisäksi asennusjonon määrittämisestä saa laitettua Windows-päivitykset asentumaan automaattisesti. Näistä syistä riittää, että asennuksessa käytettävä Windows-versio on tarpeeksi uusi. Ajurien lataamisen jälkeen ne puretaan valittuun kansioon.

Seuraavaksi käynnistetään käyttöönotto työkalu, ja avataan palvelimen F-levyllä sijaitseva käyttöönottoasema, jossa asennusjonot sijaitsevat ja josta ne välittyvät verkon

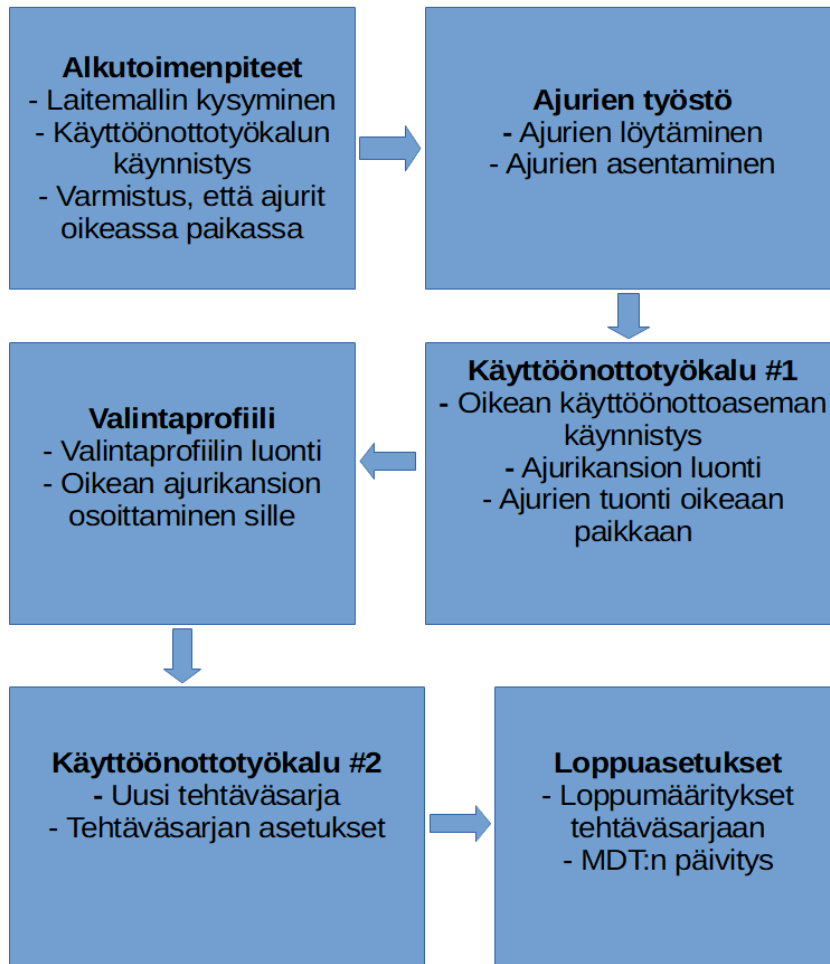
kautta asennettaville laitteille. Sinne luodaan uusi kansio, joka nimetään laitemallin mukaan, esimerkiksi HP ProBook 430 G4. Tähän kansioon tuodaan käyttöönotto työkalun tuontiominaisuudella aiemmin ladatut ja puretut ajurit. Seuraava toimenpide on uuden Selection Profilen eli valintaprofiilin luominen, jolle annetaan nimeksi laitemalli ja laitteen bittisyys. Valintaprofiilille annetaan sen jälkeen aiemmin tehty ajuripakettikansio. Tämän jälkeen luodaan uusi Task Sequence eli tehtäväsarja, jonka nimeksi tulee laitteen, järjestelmän bittisyyden ja Windowsin kielen yhdistelmä, esimerkiksi HP ProBook 445R G6 x64 FI. Tehtäväsarjassa määritellään Windows-versio ja sen kieli, organisaation nimi, järjestelmänvalvojan tunnuksen salasana sekä oikea valintaprofiili, joka sisältää ajuripaketin. Asennusjono siis rakentuu näistä kahdesta komponentista, valintaprofiilista ja tehtäväsarjasta. Lopputoimenpiteinä asetetaan palomuurin asetukset ja sallitaan Windowsin päivitykset ennen ja jälkeen asennuksen. Englannin- ja suomenkielisille versioille tehdään omat jononsa.

Muutamien kokeilujen jälkeen päätettiin, että ajuripaketti olisi hyvä ladata valmiiksi ladatut tiedostot -kansioon, josta robotti sitten hakee sen ja työstää sen oikeaan muotoon. Tämä siksi, että ensinnäkin HP:n latausohjelma tarvitsee jokaisella käynnistyskerralla järjestelmänvalvojan oikeuksia. Vaikka UiPathin käynnistäisi järjestelmänvalvojana, sillä on silti vaikeuksia päästä käsiksi sellaisiin ohjelmiin, jotka niitä tarvitsevat. Toisekseen HP:n ajuriportaalista selaimen kautta ladattuna törmättiin taas UiPathin ongelmiin löytää ilmestyviä ikkunoita selaimessa, kuten ”Haluatko varmasti tallentaa tämän tiedoston?” - ilmoituksen ajuripakettia ladattaessa. Viimeisenä syynä on se, että ajuripaketin latauksessa saattaa mennä jopa 2 tuntia, koska ne ovat tiedostokooltaan suuria. Robotille annettava lähtöasetelma on siis laitemalli, jota se kysyy itse, jonka lisäksi ajuripaketin pitää olla valmiiksi ladattu ja sen pitää löytyä uusimpana tiedostona ladatut tiedostot -kansioista. Jos robotti ei löydä EXE-päätteistä tiedostoa tuoreimpana, pyydetään käyttäjää varmistamaan, että ajurit on ladattu oikeaan paikkaan.

4.3 Toiminnallisuuden osiin jakaminen UiPathissa

Parhaiten hallittavissa ja muokattavissa olevan kokonaisuuden ohjelmaan saa jakamalla sen pienempiin osiin. UiPathissa toiminta jaetaan sekvensseihin, jotka laitetaan suorittamaan tietyssä järjestyksessä. Yksittäisen sekvenssin voi myös suorittaa erillään, jolloin

jokaista ohjelman osaa voi testata erillään muista. Tarkoituksena on, että myös asennusjonorobotin toiminnallisuus jaetaan pienempiin osiin, jotta esimerkiksi sen jatkokehittäminen ja hallinta ovat helpompaa. Osiin jakaminen tapahtuu kuvan 5 mukaisesti kuuteen osaan.



Kuva 5. Sekvenssijako. Ohjelmistorobotin toiminnallisuuden jakaminen kuuteen osaan.

Alkutoimenpiteissä kysytään laitemallin nimi, käynnistetään käyttöönottotyökalu ja varmistetaan, että ajurit on ladattu oikeaan paikkaan. Sitä seuraa ajurien työstö, jossa ajuripaketti asennetaan, eli puretaan kansioon. Tämän jälkeen siirrytään takaisin käyttöönottotyökaluun, jossa luodaan uusi ajurikansio laitemallin nimellä, jonka jälkeen siihen tuodaan ajurit. Suoritus jatkuu valintaprofiilin luomisella, josta siirrytään tehtäväsarjan tekemisen kautta loppuasetuksiin. Ohjelma on näin kompaktisti jaettu ja sen osien muokkaaminen ja jatkokehittäminen on helppoa.

5 Toteutus

Ohjelmistorobotin käytännön toteutus tehtiin Windows-palvelimella UiPath Studio -ohjelmalla. Windowsin palvelimen ja tavallisen käyttöjärjestelmän ero on se, että palvelimella ylläpidetään palveluja suurelle määrälle käyttäjiä, ja sieltä löytyy myös työkaluja tähän tarkoitukseen. Käytettävällä palvelimella sijaitsee MDT-työkalu, jolla luodaan asennusjonoja ja joka jakaa niitä asennettaville tietokoneille. Tässä osiossa kerrotaan ohjelmistorobotin toteutuksesta, haasteista ja ratkaisuista käytännön tasolla ja koodiesimerkeillä.

5.1 Alkutoimenpiteet ja ajurien työstö

Sovellusten käynnistämiseen löytyy UiPathista valmis toiminto, jota hyödynnettiin käyttöönottotyökalun käynnistämisessä. Laitemallin kysymiseen käyttäjältä käytettiin myös valmista toimintoa, joka tallettaa saamansa vastauksen string-tyyppiseen, eli tekstiä olevaan muuttuun. Kuvassa 6 on UiPathin muuttujanäkymä.

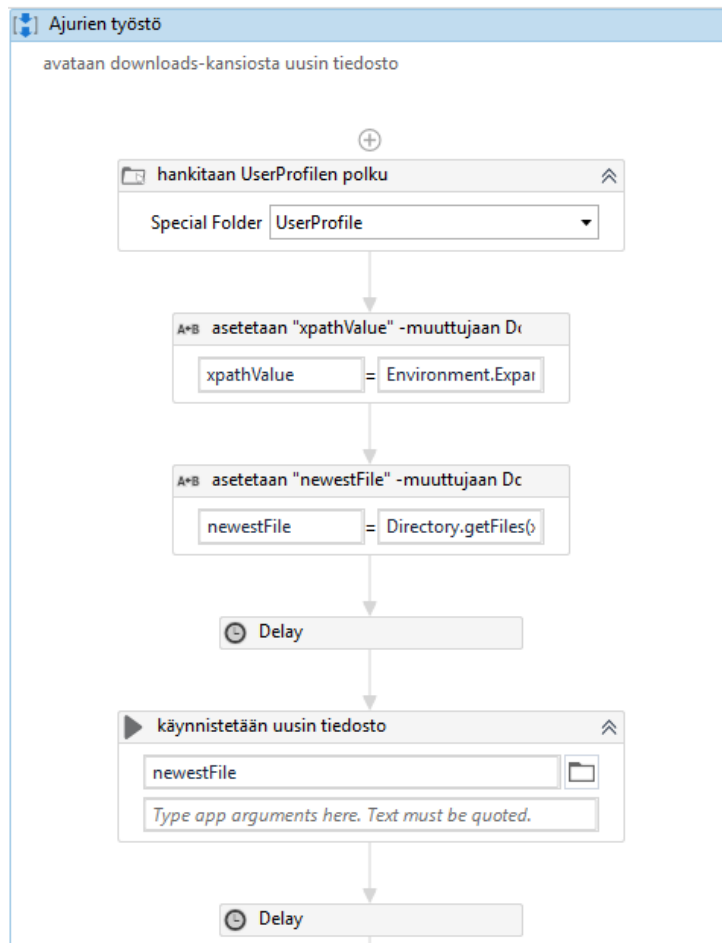
Name	Variable type	Scope	Default
laiteMalli	String	Flowchart	"ProBook x360"
ajuriMalli	String	Flowchart	"sp101495.exe"
newestFile	String	Flowchart	<i>Enter a VB expression</i>
xpathValue	String	Flowchart	<i>Enter a VB expression</i>
saveClick	String	Flowchart	<i>Enter a VB expression</i>
dsLocation	String	Flowchart	"F:\MDT\DeploymentShare"
ajuriLocation	String	Flowchart	"C:\SWSetup"

Kuva 6. UiPathin muuttujanäkymä. Vasemmalta luettuna ensin muuttujan nimi, tyyppi, toimintalue ja mahdollinen valmiiksi annettu arvo.

Muuttujille voi antaa valmiin arvon esimerkiksi testausta varten. Tässä laitemalliksi on annettu valmiiksi ProBook x360, jotta sitä ei tarvitsisi joka käynnistyksellä syöttää uudelleen.

Käyttöönottotyökalun käynnistyttyä ja laitemallin nimen saatuaan ohjelma muistuttaa käyttäjää, että ajuripaketin tulee olla ladatuissa tiedostoissa uusimpana, jotta robotti löytää sen. Tässä kohtaa törmättiin ongelmaan. Alun perin haluttiin, että robotti tunnistaa automaattisesti, onko oikea ajuripaketti oikeassa kohdassa ladatuissa tiedostoissa. Tätä varten olisi robotille pitänyt antaa ajuripaketin nimi, kuten "sp101495.exe", jota se olisi etsinyt kansioista. Kehityksessä tehtiin kuitenkin ratkaisu, että ajuripaketin nimen etsiminen ennen robotin suoritusta on työläämpää kuin pikainen varmistus siitä, että ajurit on ladattu oikeaan paikkaan. Varmistuksen jälkeen robotin suoritus siirtyy seuraavaan sekvenssiin eli ajurien työstöön.

Ensimmäinen tehtävä oli löytää ajuripaketti oikeasta kansioista. Tähän käytettiin toimintotarjua, jossa haetaan tietokoneen käyttäjän tiedostopolku ja sitä kautta ladatut tiedostot -kansion uusin tiedosto. Kuvassa 7 on tämä toiminto UiPathin koodin muodossa.



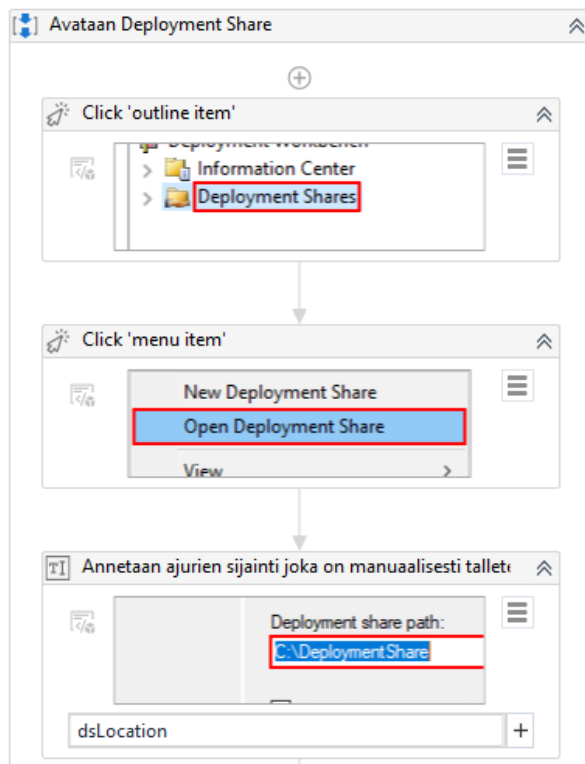
Kuva 7. Uusimman tiedoston löytäminen ladatut tiedostot -kansioista.

Kuvassa robotti hakee tietokoneen käyttäjän tiedostopolun, jota kautta se löytää ladatut tiedostot -kansion. Sen jälkeen kansion uusimman tiedoston polku asetetaan newest-File-muuttujaan, joka käynnistetään lopuksi. Uusimman tiedoston löytäminen tietystä kansioista muistuttaa paljolti tavallista ohjelmointia, sillä esimerkiksi funktio `Directory.GetFiles()` piti kirjoittaa tavallisena tekstipohjaisena koodina. Näissä tilanteissa ohjelmointitaustasta on huomattava apu UiPathin käyttämiseen.

Ajuripaketti käynnistyy asennusikkunaan, koska se on sovellus, joka purkaa omat tiedostonsa haluttuun paikkaan. Purkaminen tapahtuu robotille näytetyillä klikkauskomendoilla. Purkukansio on aina sama, joten se on valmiiksi laitettu muuttujan arvoksi, jota myöhemmin käytetään ajurien löytämiseen. Jatkokehitystä ajatellen kansiota olisi hyvä vaihtaa, koska nyt se pitää suorituksen jälkeen poistaa, jotta seuraavalla kerralla ei käytettäisi samoja ajureita uuden laitemallin kanssa. Kun ajurit on purettu, ollaan valmiita siirtymään seuraavaan sekvenssiin.

5.2 Käyttöönottotyökalun ensimmäinen toiminto ja valintaprofiili

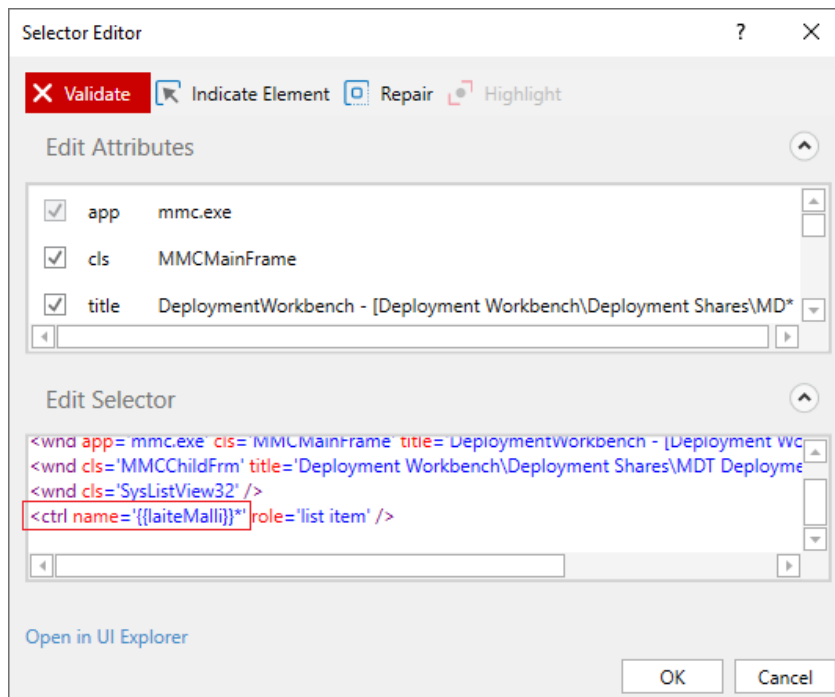
Seuraavana toimenpiteenä on käynnistää käyttöönottotyökalussa oikea käyttöönotto-asema, johon asennusjono halutaan luoda. Käyttöönottoaseman tiedostopolkua kysyttäessä hyödynnettiin UiPathin `Type Into` -toimintoa, joka kirjoittaa haluttuun kenttään halutun tekstin, esimerkiksi muuttujaan tallennetun tekstijonon. Käyttöönottoaseman tiedostopolku tallennettiin tätä varten muuttujaan. Muuttujan arvoa voi helposti muuttaa, jos aseman sijainti vaihtuu. Kuva 8 havainnollistaa, miltä tämä toiminto näyttää UiPathissa.



Kuva 8. Käyttöönottoaseman avaus UiPathissa.

Aluksi robotille näytetään, mistä sen tulee klikata, jotta päästään syöttämään käyttöönottoaseman oikea sijainti. Ensimmäinen klikkaus on määritetty klikattavaksi hiiren oikealla nappulalla, jotta seuraavan kohdan listaus avautuu. Listauksesta robotti etsii käyttöönottoaseman avauksen, johon se kirjoittaa muuttujaan tallennetun tiedostopolun. Muuttujan nimi on tässä tapauksessa dsLocation. Käyttöönottoaseman avauduttua voidaan siirtyä ajurikansion tekemiseen.

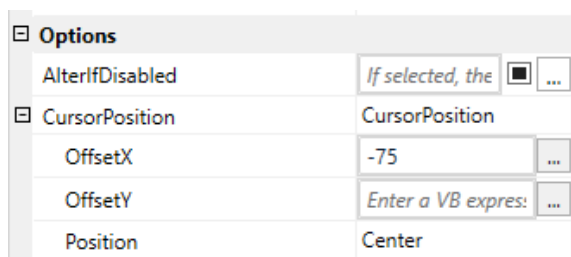
Ajurikansio täytyy myös luoda oikeaan paikkaan. Sinne pystyttiin navigoimaan pelkillä klikkauskomennoilla, koska käyttöönottoasema ja sen tiedostopuu on nyt avattuna. Kun robotti löytää oikean sijainnin, se luo sinne kansion laitemallin muuttujaan talletetulla nimellä. Kansio pitää seuraavaksi löytää, jotta sinne voidaan tuoda ajurit. Käyttöönottotyökalussa ei pystynyt lajittelemaan kansioita, joten tässä kohtaa piti turvautua kansion löytämiseen nimen perusteella. UiPathissa on ominaisuus, jossa esimerkiksi kansiota voidaan etsiä jollain tietyllä nimellä. Koska robotti loi kansion laitemallin nimellä, voidaan samaa muuttujaa käyttää myös kansion etsimiseen. Kuva 9 havainnollistaa tätä ominaisuutta.



Kuva 9. Kohteen etsiminen tietyllä nimellä UiPathissa.

Kuvassa näkyy valintaeditori, jossa robotille näytettyä toimintoa voi muokata perusteellisemmin. Punaisella suorakulmiolla on rajattu kohta, jossa on halutun kohteen nimi. Tässä siihen on laitettu laiteMalli-muuttuja, johon on tekstimuotoisena tallennettu asennusjonon laitemallin nimi. Perässä oleva tähtimerkki tarkoittaa, että laitemallin nimen jälkeen voi tulla mitä tahansa merkkejä. Jos nimen perään jää esimerkiksi välilyönti, löytyy kohde silti. Näin löydetään haluttu kohde ilman lajitteluominaisuuksia. Kun robotti on löytänyt luomansa kansion, se tuo sinne oikeat ajurit klikkaamalla Import Drivers -nappulaa ja antamalla ajurien sijainnin eli tiedostopolun, joka on tallennettu muuttujaan.

Valintaprofiilin luonti on hyvin samanlainen prosessi kuin ajurikansion luonti. Valintaprofiilin kansio luodaan oikeaan paikkaan, joka onnistuu pelkillä klikkauskomennoilla. Valintaprofiilin nimeksi tulee laitemalli ja x64-laitteen bittisyyden mukaan. Ajurikansiota osoitettaessa törmättiin pieneen ongelmaan, jossa UiPath ei tunnistanut valintaruutua, joka olisi pitänyt ruksia oikean ajurikansion valitsemiseksi. Tässä kohtaa turvauduttiin offset-ominaisuuteen, joka siirtää osoitettua klikkausta tietyn pikselimäärän verran johonkin suuntaan. Tämä on melko epätarkka menetelmä, joka todennäköisesti menee rikki esimerkiksi näytön koon muuttuessa. Kuvassa 10 näkyy tämä toiminto.

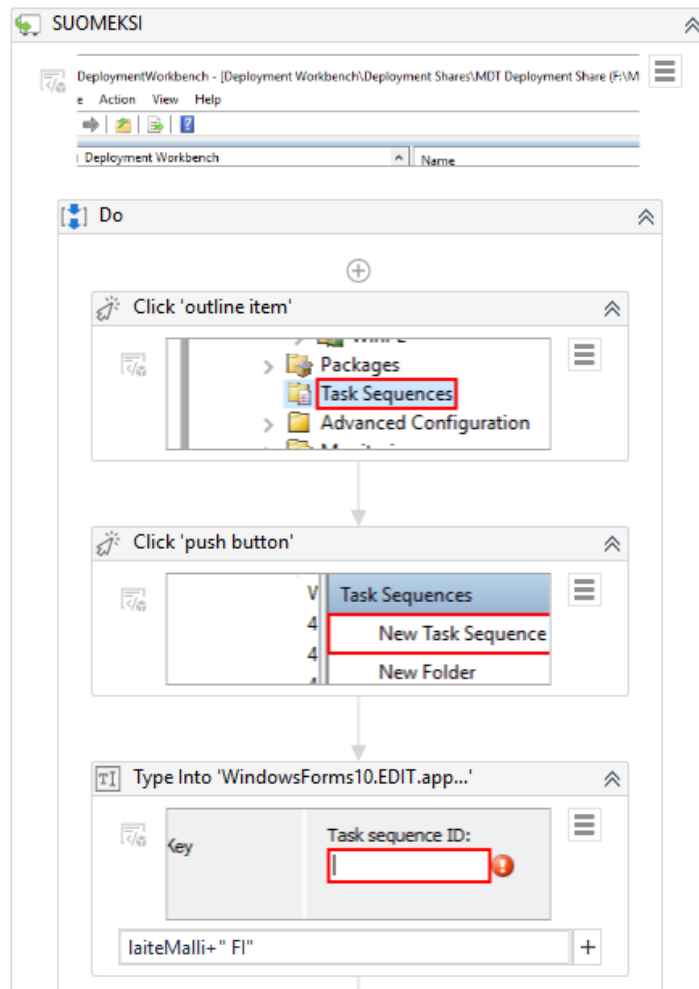


Kuva 10. Klikkauksen lisäasetukset. Tässä on turvauduttu offset-toimintoon, joka siirtää klikkausta 75 pikseliä vasemmalle.

Kun klikkaus siirtyy 75 pikseliä vasemmalle, se osuu vihdoin valintaruutuun, ja valintaprofiili saa oikean ajurikansion. Tämän jälkeen valintaprofiili on valmis, ja robotti voi siirtyä eteenpäin.

5.3 Tehtäväsarjat ja lopputoimenpiteet

Tehtäväsarjassa asennusjonolle osoitetaan oikea käyttöjärjestelmä ja alkuun muutama perusasetus, kuten järjestelmänvalvojan tunnus ja salasana. Suomen- ja englanninkielisille versioille tehdään omat tehtäväsarjansa, koska niille annetaan eri käyttöjärjestelmät kielen mukaan. Nämä toimenpiteet ovat hyvin virtaviivaisia ja onnistuivat pelkillä klikkauskomennoilla käyttöönotto työkalun sisällä. Tehtäväsarjojen nimeämisessä hyödynnettiin taas laiteMalli-muuttujaa, joka sisältää tekstinä laitemallin nimen. Kuva 11 havainnollistaa tehtäväsarjan luomista.



Kuva 11. Tehtäväsarjan luonti.

Ohjelmistorobotti kiinnittyy aluksi käyttöönotto työkalun ikkunaan, jonka jälkeen se klikkailee tiensä läpi tehtäväsarjan luonnin. Suomenkielinen tehtäväsarja luodaan ensimmäiseksi, jonka jälkeen englanninkielinen. Varsinainen ero näillä kahdella on eri käyttöjärjestelmä ja nimi.

Kun tehtäväsarjat on luotu, asetetaan molempiin vielä loppuasetukset. Loppuasetuksiin lukeutuu muun muassa komentosarja palomuurin asetuksista, jotka on tallennettu muuttujaan sekä Windowsin päivitysten salliminen ennen ja jälkeen asennuksen. Tämän jälkeen robotti vielä päivittää MDT:n, jotta uusin asennusjono tulee näkyviin.

5.4 Testaus

UiPathilla testaaminen osoittautui melko hankalaksi. Varsinaista testaamista varten olisi pitänyt ladata yhteisön luoma ohjelmistokehys eli framework. Tässä kohtaan puntaroitiin, kuinka mielekästä ja hyödyllistä se on, ja päätettiin jatkaa manuaalisella testauksella. Robottia siis testattiin vaiheittain sen luomisen aikana, koska ohjelmistorobotille pitää muutenkin näyttää koko prosessi, jota sille ohjelmoidaan. Vaihe vaiheelta testattu robotti toimii, mutta jos esimerkiksi prosessiin tulee muutoksia, ei robotin toimintakykyä voi selvittää testeillä, vaan se pitää suorittaa kokonaan tai osissa. UiPathin tulevaisuutta ajatellen sen olisikin hyvä kehittää jotain mielekkäämpää testaustapaa, vaikka ohjelmistorobotin testaus onkin lähtökohtaisesti haastavaa.

6 Tulokset

Tuloksena insinööriyöstä syntyi toimiva ohjelmistorobotti ja selvitys ohjelmistorobotiikasta, sekä automaatiokohteiden valinnasta ja hyvän automaation käytänteistä. Markkinoiden johtava ohjelmistorobotiikan työkalu UiPath tuli myös tutuksi, ja sen hyviä ja huonoja puolia sekä eroja kilpailijoihin arvioitiin.

Ohjelmistorobotin varsinaisten testien puute saattaa olla ongelma jatkokehitystä ajatellen. Robotti on kuitenkin erittäin altis käyttöliittymien muutoksille, ja muutama siihen ohjelmoitu toimenpide on todella herkkä ja voi rikkoutua helposti. Mikäli UiPathiin tulee tulevaisuudessa parempia testausmenetelmiä, olisi niiden implementointi jatkokehityksen kannalta ensimmäinen toimenpide.

Konkreettisesti robotti säästää tällä hetkellä 20-30 työtuntia vuodessa, mutta se voi tulevaisuudessa säästää enemmänkin, jos asennusjonojen luomisen kynnyks madaltuu. Käsin tehtynä asennusjonon rakentamiseen saattaa vierähtää jopa kokonainen työpäivä muiden tehtävien ohessa, kun taas robotti hoitaa sen itsenäisesti alle tunnissa. Robotti otetaan varsinaiseen käyttöön jo lähiaikoina, kun Lähetysseura uusii laitteistoaan ja uusille konemalleille pitää tehdä asennusjonot.

7 Johtopäätökset

Ohjelmistorobotiikka tulee olemaan tulevaisuudessa isossa roolissa, kun työtehtäviä automatisoidaan. En usko, että se tulee varsinaisesti viemään työpaikkoja ihmisiltä, vaan se enemmänkin helpottaa ihmisten työtä hoitamalla tylsät rutiinitehtävät. Tekoäly ja koneoppiminen tulevat varmasti olemaan vahvasti läsnä ohjelmistorobotiikan tulevaisuudessa, kun robotit alkavat työstämään yhä monimutkaisempia tehtäviä.

Automaatiokohteita valittaessa olisi hyvä kiinnittää kokonaisvaltaista huomiota automatisoitavaan prosessiin. Kaikkia prosesseja ei tarvitse automatisoida, mutta ne, jotka halutaan automatisoida, on hyvä käydä perusteellisesti läpi ja selvittää, mitä kaikkia järjestelmiä ja teknologioita ohjelmistorobotin kehittämisessä tarvitsee ottaa huomioon.

Ohjelmistorobotin kehitystyöhön ei tarvitse laajaa kokemusta tai alan tutkintoa, mutta ne auttavat huomattavasti. Ohjelmistorobotiikan työkalut ovat hyvin samanlaisia kuin tavalliset kehitysympäristöt, joten niiden tunteminen helpottaa ohjelmoinnin aloittamista.

Lähteet

- 1 Ostdick, Nick. The Evolution of Robotic Process Automation (RPA): Past, Present, and Future. Verkkoaineisto. <<https://www.uipath.com/blog/the-evolution-of-rpa-past-present-and-future>> Luettu 27.1.2020.
- 2 Willcocks, Leslie P. & Lacity, Mary C. 2015. Businesses will increasingly use robots to deal with the explosion of data. Verkkoaineisto. <<https://blogs.lse.ac.uk/businessreview/2015/09/15/businesses-will-increasingly-use-robots-to-deal-with-the-explosion-of-data/>> Luettu 2.3.2020.
- 3 Mitä on ohjelmistorobotiikka? 2019. Verkkoaineisto. <<https://staria.com/fi/blogi/mita-ohjelmistorobotiikka/>> Luettu 30.1.2020.
- 4 Boulton, Clint. 2018. What is RPA? A revolution in business process automation. Verkkoaineisto. <<https://www.cio.com/article/3236451/what-is-rpa-robotic-process-automation-explained.html>> Luettu 10.2.2020.
- 5 Efima Oy. 2019. Robotiikka ja tekoäly taloushallinnossa.
- 6 UiPath Named a Leader in Forrester Wave for Robotic Process Automation, Q4 2019. 2019. Verkkoaineisto. <<https://www.uipath.com/blog/uipath-2019-industry-leader-forrester-wave-rpa>> Luettu 5.4.2020.
- 7 UiPath Named a Leader in the 2019 Gartner Magic Quadrant for Robotic Process Automation Software. 2019. Verkkoaineisto. <https://www.uipath.com/blog/uipath-2019-industry-leader-gartner-rpa-magic-quadrant> Luettu 5.4.2020.