



Rasmus Julin

Liiketoimintaprosessin automatisointi ohjelmistorobotilla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

26.11.2021

Tiivistelmä

Tekijä: Rasmus Julin
Otsikko: Liiketoimintaprosessin automatisointi ohjelmistorobotilla
Sivumäärä: 34 sivua
Aika: 26.11.2021

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintäteknikka
Ammatillinen pääaine: Ohjelmistotuotanto
Ohjaajat: Lehtori Simo Silander

Opinnäytetyön tarkoituksena oli tutkia ohjelmistorobotin roolia liiketoimintaprosessin automatisoinnissa, robotin kehitysprosessia ja tarkastella olemassa olevia työkaluja. Tarkoituksena oli myös käydä läpi ohjelmistorobotiikan mahdollisia ongelmia ja miettiä robotiikan tulevaisuutta liiketoimintaprosessien automatisoinnissa.

Työssä käydään läpi, miten ohjelmistorobotiikkaa hyödynnetään liiketoimintaprosessin automatisoinnissa ja vertaillaan ohjelmistorobotiikan keskeisiä teknologioita. Tarkastellaan ohjelmistorobotiikan hyötyjä ja rajoitteita ja käydään myös lyhyt katsaus ohjelmistorobotiikan historiaan. Työssä pohditaan myös, millaiseen suuntaan ohjelmistorobotiikka tulevaisuudessa kehittyy ja millainen rooli tekoälyllä tässä on. Työssä esitellään todelliseen toteutukseen pohjautuva esimerkkirobotti prosessikaavioilla.

Lopputuloksena työssä saatiin esitettyä kattava kuva ohjelmistorobotiikasta ja siitä, miten sitä voidaan hyödyntää liiketoimintaprosessien automatisoinnissa. Työtä voi myös käyttää ohjelmistorobotiikan teknologian valinnan tukena.

Avainsanat: Ohjelmistorobotiikka, RPA, Blue Prism, Python, Robot Framework, Tekoäly

Abstract

Author: Rasmus Julin
Title: Automating Business Process with Software Robot
Number of Pages: 34 pages
Date: 26 November 2021

Degree: Bachelor of Engineering
Degree Programme: Information and Communication Technology
Professional Major: Software Engineering
Instructors: Simo Silander, Senior Lecturer

The purpose of the final project was to study business process automation with a software robot and compare the main technologies in robotic process automation. One of the main objectives was also to investigate the benefits and the restrictions in robotic process automation and think about the future of the industry.

The thesis focuses on robotic process automation and how to utilize it on business process automation. There is also a brief review of the history of software robotics. The differences on the main technologies were thoroughly studied and compared. This thesis also contains an example of a software robot and its process. The future of the software robotics and artificial intelligence is also reviewed.

The final product is a comprehensive study about the robotic process automation and how it can be utilized in automating business processes. This thesis can be also used as a tool to help choosing the right robotic process automation technology.

Keywords: Robotic Process Automation, Blue Prism, Python, Robot Framework, Artificial intelligence

Sisällys

Lyhenteet

1	Johdanto	1
2	Ohjelmistorobotiikka	2
2.1	Mitä on ohjelmistorobotiikka?	2
2.2	Historia	4
2.3	Yleisimmät ohjelmistorobotiikan työkalut ja teknologiat	5
2.3.1	Blue Prism	6
2.3.2	Python	12
2.3.3	Robot Framework	14
2.4	Hyödyt	16
2.5	Rajoitteet	16
3	Automatisoitava prosessi	19
3.1	Liiketoimintaprosessi	19
3.2	Datan laatu	20
4	Esimerkkirobotti	22
4.1	Kokonaisprosessi	22
4.2	Robotin prosessi	23
5	Ohjelmistorobotiikka tulevaisuudessa	27
5.1	Tekoäly ja koneoppiminen	27
5.2	Hyperautomaatio	29
6	Yhteenveto	31
	Lähteet	32

Lyhenteet

- RPA: *Robotic Process Automation*. Teknologia rutiininomaisten työtehtävien automatisointiin erilaisissa prosesseissa.
- RDP: *Remote Desktop Protocol*. Etätyöpöytäprotokolla, jonka avulla pystyy käyttämään tietokonetta etänä verkkoyhteyden läpi.
- OCR: *Optical Character Recognition*. Teknologia, jonka avulla voidaan tunnistaa tekstiä sähköisesti muokattavaan muotoon.
- AI: *Artificial Intelligence*. Koneen tai ohjelman kyky jäljitellä ihmismäistä päättelyä.
- ML: *Machine Learning*. Tekoälyn yksi osa-alue, jonka avulla sovellus voi tehdä uusia päättelyitä tehtyjen toimintojen perusteella.
- HTML: *Hypertext Markup Language*. Verkkosivujen standardoitu merkintäkieli.
- XPath: *XML Path Language*. Kyselykieli XML-pohjaisten dokumenttien osittamiseen.
- SQL: *Structured Query Language*. Standardoitu kyselykieli tietokantojen datan hallintaan ja käsittelyyn.
- API: *Application Programming Interface*. Ohjelmointirajapinta, jonka avulla sovellukset kommunikoivat keskenään.

1 Johdanto

Maailman digitalisoituessa ja tietotekniikan kehittyessä yritysmaailman vanhat paperityöt tehdään yhä useammin digitaalisesti. Digitaalinen työympäristö ei kuitenkaan tarkoita sitä, että uuvuttava ja toistuva rutiinityö olisi poissa. Rutiininomaisen työn luonne on vain muuttunut. Sähköisten hakemusten käsittely, tiedon syöttäminen järjestelmiin, sähköpostien käsittely ja lajittelu, datan validointi ja muut toistuvat tehtävät hallitsevat monen yrityksen arkea ja vievät paljon työaika-asiiantuntijoilta, mitä voisi käyttää liiketoiminnan kannalta tärkeämpiin tehtäviin. Jatkuva digitaalinen kehitys tuo myös mukanaan uusia järjestelmiä, integraatioita ja data-alustoja, jolloin on tarve usein siirtää dataa vanhasta järjestelmästä uuteen. Näissä tilanteissa ratkaisuna voisi olla ihmistä imitoiva ohjelma eli ohjelmistorobotti, joka on opetettu esimerkiksi siirtämään dataa järjestelmien välillä.

Opinnäytetyön tarkoituksena on tutkia ohjelmistorobotin roolia liiketoimintaprosessin automatisoinnissa, robotin kehitysprosessia ja vertailla olemassa olevia työkaluja, käydä läpi ohjelmistorobotiikan mahdollisia ongelmia ja miettiä robotiikan tulevaisuutta liiketoimintaprosessien automatisoinnissa. Kirjoittaja on työskennellyt ohjelmistorobotiikan parissa useamman vuoden ja ollut automatisoimassa toistakymmentä liiketoimintaprosessia ohjelmistorobotiikalla useammalla eri teknologialla ja työkalulla.

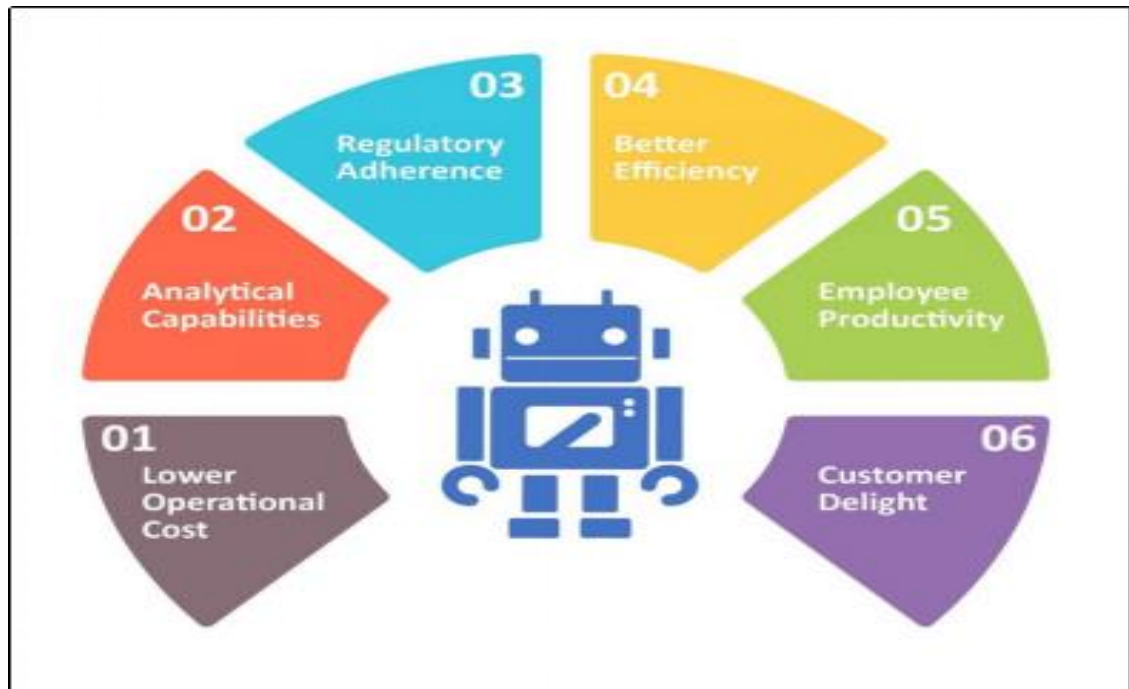
2 Ohjelmistorobotiikka

2.1 Mitä on ohjelmistorobotiikka?

Nykyään mediassa ja yrityksissä puhutaan yhä enemmän roboteista ja robotiikasta, mutta mitä sillä oikeasti tarkoitetaan? Ohjelmistorobotiikka eli RPA (Robotic Process Automation) tarkoittaa teknologiaa, jolla voidaan automatisoida säännönmukaisia työtehtäviä. [1.]

Ohjelmistorobotti ei siis ole mekaaninen kone, joka suunnittelee ajavansa ihmiset työttömäksi, vaan ohjelmisto, jota voidaan hyödyntää säännönmukaisten tehtävien suorittamiseen. Valtavien massojen käsittelyyn liittyvä manuaalinen työ aiheuttaa ihmisiltä väkisinkin inhimillisiä virheitä ja pahimmissa skenaarioissa työntekijän työuupumusta ja loppuunpalamisen. Isoissa massoissa inhimilliset virheet tulevat usein yritykselle kalliiksi ja aiheuttavat pitkälle meneviä virheselvityksiä, kun virheellistä dataa pääsee useampaan eri järjestelmään. Tämä saattaa näkyä suoraan myös asiakkaan puolella, jos asiakasviestit, asiakastiedot tai laskut lähtevät väärille asiakkaille. Tämän virhemarginaalin minimoimiseksi ohjelmistorobotti on täydellinen kumppani.

Kuvassa 1 esitetään, miten robotti voi olla hyödyksi yritykselle ja mitä ominaisuuksia se tuo. Sen avulla voidaan parantaa tehokkuutta, pienentää kustannuksia ja parantaa työntekijöiden tuottavuutta.



Kuva 1. Ohjelmistorobotin ominaisuuksia [2]

Robotti on ohjelma, joka mallinnetaan toimimaan käyttöliittymällä ihmisten taivoin. Se on parhaimmillaan, kun kyseessä on rutiininomainen tehtävä, mikä vaatii paljon toistoja ja huolellisuutta. Robotit hyödyntävät ennalta määriteltyjä sääntöjä ja pystyvät käyttämään järjestelmiä pitkälti samaan tapaan kuin ihminen näitä käyttäisi, ilman inhimillisiä virheitä tai työuupumusta [1]. Sen avulla voidaan vapauttaa työntekijät tekemään vaativampia ja miellyttävämpiä tehtäviä, jotka tuovat liiketoiminnalle enemmän arvoa. Kuvassa 2 esitetään Digital Workforcen tekemän tutkimuksen tulokset. Tutkimuksessa ihminen ja robotti syöttää saman datan Excel-tiedostosta järjestelmään ja verrataan, miten tehty työ samassa ajassa suhteutuu.

	Ihminen	Ohjelmistorobotti
Syötetyt tiedot (kpl)	20	20
Käytetty aika	9min 9sek	33sek

Kuva 2. Digital Workforcen tutkimustulokset ihmisen ja robotin eroista samassa prosessissa [3]

Konkreettinen esimerkki ohjelmistorobotin tehtävästä voisi olla, kun tarvitaan datan siirtämistä järjestelmästä toiseen, mutta perinteiset integraatiot ovat järjestelmien osalta mahdottomia ja tämä pitäisi tehdä käsin. Useita ihmisiä vaativa työ voitaisiin siirtää robotille, joka ihmisen tapaan hakisi lähdejärjestelmästä tarvittavat tiedot ja kirjaisi ne kohdejärjestelmään.

Ohjelmistorobotiikkaa tehdään nykyisin useilla erilaisilla työkaluilla ja teknologioilla, ja ne kehittyvät nopeaa tahtia. Osa työkaluista on valmiita sovelluksia, joissa robotti luodaan pitkälti käyttöliittymätunnistuksella ja prosessikaavioilla, kun taas toiset ovat enemmän avoimen lähdekoodin ohjelmistokehyksiä. Suosituimmat työkalut sisältävät useita erilaisia teknologioita, joiden avulla automaatiota pystytään tekemään mahdollisimman monipuolisesti, kuten esimerkiksi kuvantunnistuksella [5]. Kuvantunnistusta voidaan esimerkiksi hyödyntää, kun automatisoitava sovellus on RDP-etäyhteyden (Remote Desktop Protocol) takana, jolloin robotti ei välttämättä suoraan pääse sovelluksen taustakoodiin käsiksi. Tällöin robotille voidaan opettaa kuvia, mitä se tunnistaa taustakoodin sijasta. OCR- eli tekstintunnistusteknologiaa hyödyntämällä robotti kykenee myös lukemaan tekstiä kuvista.

Yhteinen tekijä eri työkaluissa ja teknologioissa kuitenkin on, että robottiohjelman tekeminen olisi mahdollisimman kevyttä verrattuna perinteisiin integraatioihin ja automaattioratkaisuihin. Ohjelmistorobotiikka on yleisesti myös hyvin yleiskäyttöistä ja toimialariippumatonta [4].

2.2 Historia

Ohjelmistorobotiikan juuria voidaan pitää jo vuodessa 1959, jolloin alan pioneeri Arthur Samuel kehitti koneoppimista työskennellessään tietokoneyhtiössä IBM. Koneoppinen eli Machine Learning (ML) syntyi tieteellisestä pyrkimyksestä luoda tekoälyä. Tämän avulla tietokoneilla pystyttiin tekemään uudenlaisia tekstin käännöksiä ja yhteenvetoja. [6.]

Yksi isoimmista ohjelmistorobotiikan kehitysaskelista tapahtui kuitenkin 90-luvulla, kun keksittiin datan parsiminen erilaisista sovelluksista ja internet -sivustoista. Tämä on nykypäivänäkin yksi ohjelmistorobotiikan ydinalueista, kun tunnistetaan robotille erilaisia objekteja niin web-pohjaisista järjestelmistä kuin työpöytäsovelluksesta. Samalla vuosikymmenellä ilmestyi myös ensimmäisiä työkulkuautomaatiotyökaluja ja tekoäly alkoi kehittymään. [6.]

Ohjelmistorobotiikan todellinen synty sijoittuu 2000-luvun alkuun, mutta suosio ja kehitys jäi vielä varsin suppeaksi. Vasta vuonna 2015 suosio lähti kasvamaan ja kehitys kiihtymään. [6.]

2.3 Yleisimmät ohjelmistorobotiikan työkalut ja teknologiat

Ohjelmistorobotteja voidaan kehittää nykyisin useilla erilaisilla työkaluilla ja teknologioilla. Kehitys voi tapahtua suoraan valmiilla ohjelmalla, jossa kehityksen pystyy toteuttamaan hyvin pienellä koodiosaamisella valmiiden toimintojen avulla. Näitä ovat muun muassa Blue Prism, UiPath, Automation Anywhere ja uusimpana tulokkaana Power Automate.

Robotin voi myös toteuttaa suoraan haluamallaan ohjelmointikielellä, jolloin kehitys ei ole valmiin ohjelman skaalautuvuuden varassa, vaan kehittäjällä on vapaat kädet oman osaamisensa rajoissa. Yksi tällaisista ohjelmointikielistä on Python, joka on suuressa suosiossa automaatio- ja ohjelmistorobotiikkatoteutuksissa sen dynaamisuuden ja helppolukuisen syntaksin ansiosta.

Kolmas vaihtoehto on aiemmin mainittujen hybridiratkaisu, jossa kehitys toteutetaan valmiilla sovelluskehityksellä (framework), joka on välimuoto valmiista komponenteista ja koodauksesta. Tähän on hyvä esimerkki Suomessa kehitetty Python-ohjelmointikieleen pohjautuva avoimen lähdekoodin Robot Framework, jossa robotti rakennetaan ohjelmointiympäristössä hyödyntäen Robot Frameworkin valmiita toiminnallisuuksia.

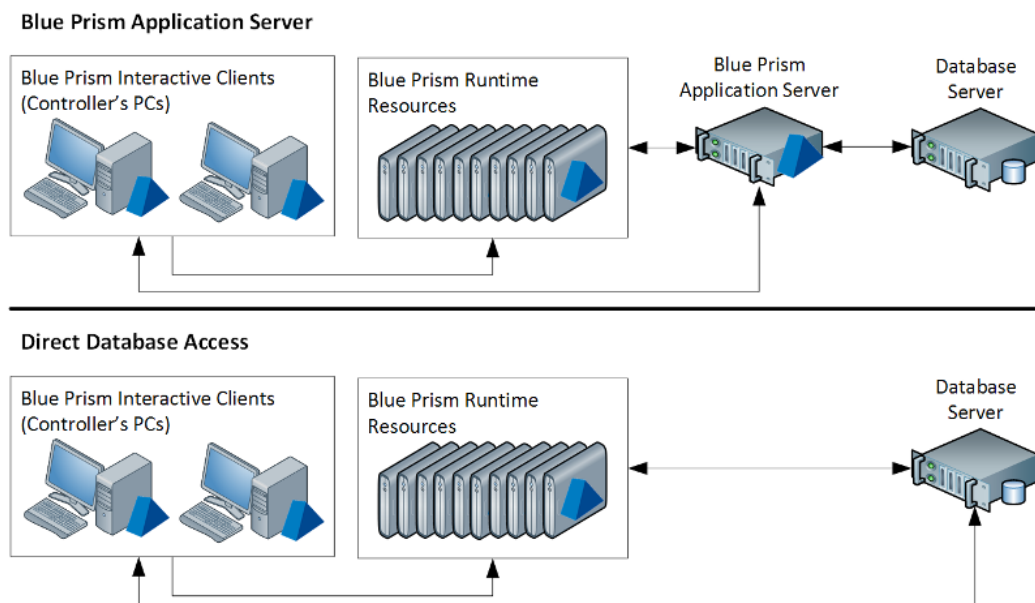
Robotin kehitystä varten työkalun valinnassa on tärkeää miettiä työkalu tarpeen mukaan, sillä jokainen työkalu on erilainen ja tämä näkyy suoraan myös työkalujen hinnassa. Valmiit käyttöliittymäpohjaiset työkalut seisovat usein lisenssihinnoittelun takana, kun taas ohjelmointikielellä itseluodut tai avoimen lähdekoodin Robot Framework -robotit ovat ilmaisia. Turhaan lisenssistä ei kuitenkaan joudu maksaa, sillä maksulliset työkalut tuovat mukanaan paljon hyviä valmiita ominaisuuksia, kuten helppotoimisen käyttöliittymän, kuvantunnistuksen, datan salauksen, valmiita tekoälytyökaluja, työjonohallinnan ja paljon muuta. Työssä käydään läpi esimerkkityökalu ja -teknologia jokaisesta edellä mainitusta kategoriasta. [7.]

2.3.1 Blue Prism

Blue Prism on yksi ohjelmistorobotiikan johtavista yrityksistä ja teknologioista. Se on etenkin Euroopassa suuressa suosiossa suuremmissa organisaatioissa, joiden tarpeisiin se on kehitetty. Blue Prism yrityksenä perustettiin Englannissa vuonna 2001 tarkoituksena kehittää uudenlaista automaatiota, joka nykyisin tunnetaan paremmin termillä RPA. [8.]

Blue Prism on visuaalisella käyttöliittymällä toimiva teknologia, jonka ansioistensa käyttö ja opettelu onnistuu pienemmällä kynnyksellä myös niille, joilla ei ole ohjelmointikielistä aiempaa tuntemusta. Se pohjautuu Microsoftin .NET-sovel- luskehukseen, johon pohjautuvat myös monet muut valmiit RPA-työkalut. [8.]

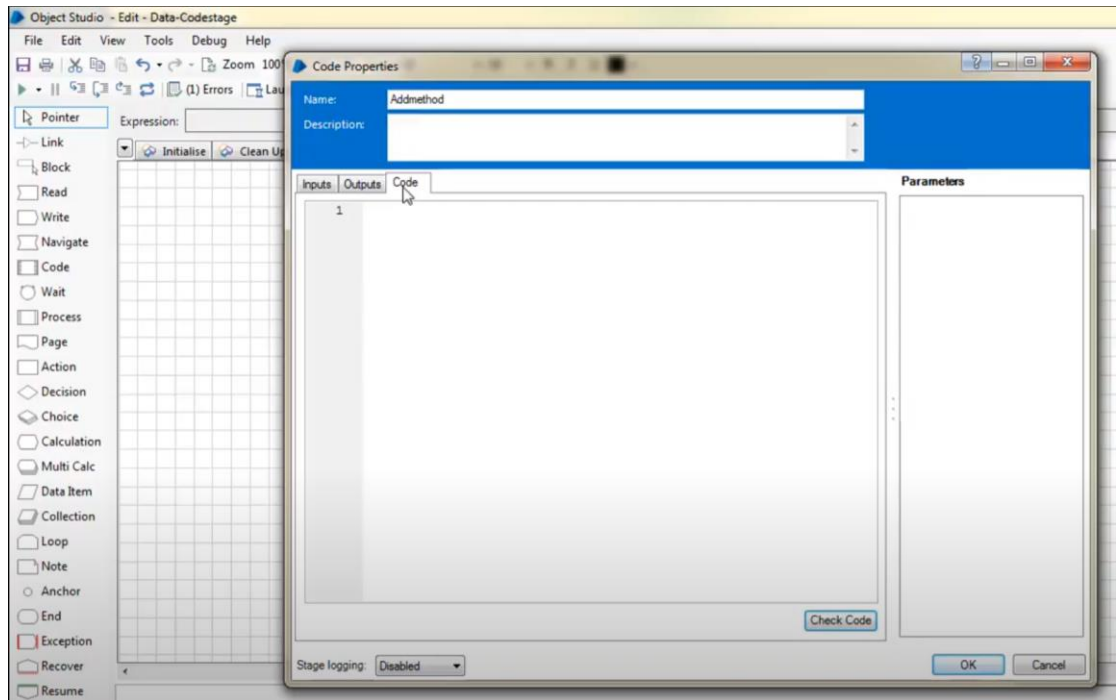
Blue Prism -sovellusympäristö on luotu valmiiksi hyvin skaalautuvaksi, ja se on mahdollista perustaa pilveen tai suoraan on-premise-ratkaisuna. Resurssiko- neet voivat myös olla palvelimia tai työpöytäkoneita [10]. Kuvassa 3 näkyy, että Blue Prism -ympäristön voi luoda suoralla yhteydellä tietokantaan tai hyödyntää erillistä Application-palvelinta resurssien hallintaan.



Kuva 3. Kaksi erilaista Blue Prism -ympäristöä [10]

Blue Prism -sovellus on itsessään valmis ja monipuolinen robottikehityksen ympäristö, missä on valmiit työkalut ja funktiot robottien rakentamiseen. Se sisältää muun muassa valmiin työjonohallinnan, tietokannan luonnin, kannan ja salasanojen salauksen, ympäristömuuttujien hallinnan, salasanojen hallinnan, kuvantunnistuksen, sovellusten elementtien tunnistuksen, versiohallinnan, login-agentin ja lokienhallintatoiminnot. Yksi hyödyllisimmistä ominaisuuksista on Blue Prism Control Room -alusta, jossa kehittäjät pystyvät ajastamaan robotteja ja seuraamaan niiden ajoa reaaliajassa. [11.]

Mikäli valmiit funktiot eivät riitä robotin toteutukseen, niin näitä on myös saatavilla Blue Prism Digital Exchange -sivulta, jossa on myös kolmannen osapuolen tekemiä ratkaisuja, mutta nämä ovat usein maksullisia. Aiemmin mainittu ohjelmointikielen osaaminen tulee kuitenkin tarpeelliseksi, jos tarvitsee luoda uusia toiminnallisuuksia itse. Blue Prism tukee C#- ja Visual Basic.NET -ohjelmointikieliä toiminnallisuuksien luontiin. Näistä yleisemmin käytössä on Visual Basic.NET, jolla lähes kaikki sovelluksen valmiit funktiot on luotu [11]. Kuvassa 4 näkyy, miten funktioita voi luoda itse "Code"-elementin avulla, jossa on sisäänrakennettu tekstieditori ohjelmointiin.



Kuva 4. Blue Prism -tekstieditori ohjelmointiin.

Ohjelmointitaustalla olevien kehittäjien on myös helpompi sisäistää itse robottien kehitystä, sillä erilaisten funktioiden, muuttujien ja looppien käyttö on ennestään tuttua. Myös sovelluksen oma syntaksi datan muunnoksille ja erilaisille päättelyille ”Calculation”-komponentin sisällä on tuttua ohjelmointitaustaisille kehittäjille, sillä logiikka ja syntaksi on hyvin samankaltainen kuin monissa ohjelmointikielissä. Kuvassa 5 on esimerkki saman toiminnon tekemisestä Blue Prismillä, Pythonilla ja Javalla.

Java

```
str1.contains(substring)
```

Python

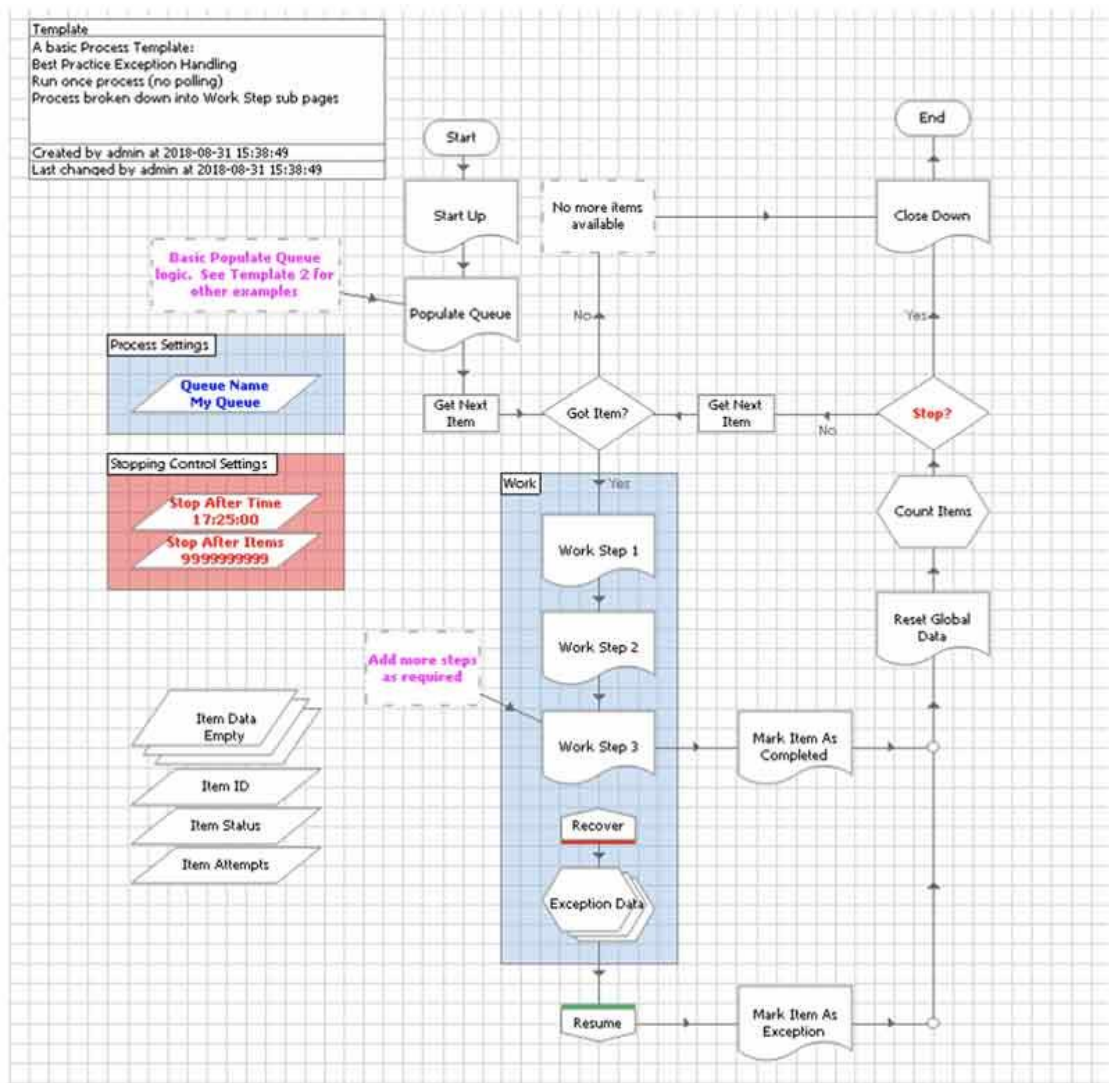
```
str1.find(substring)
```

Blue Prism

```
InStr([str1];[substring])
```

Kuva 5. Sama toiminnallisuus Javassa, Pythonissa ja Blue Prismissä

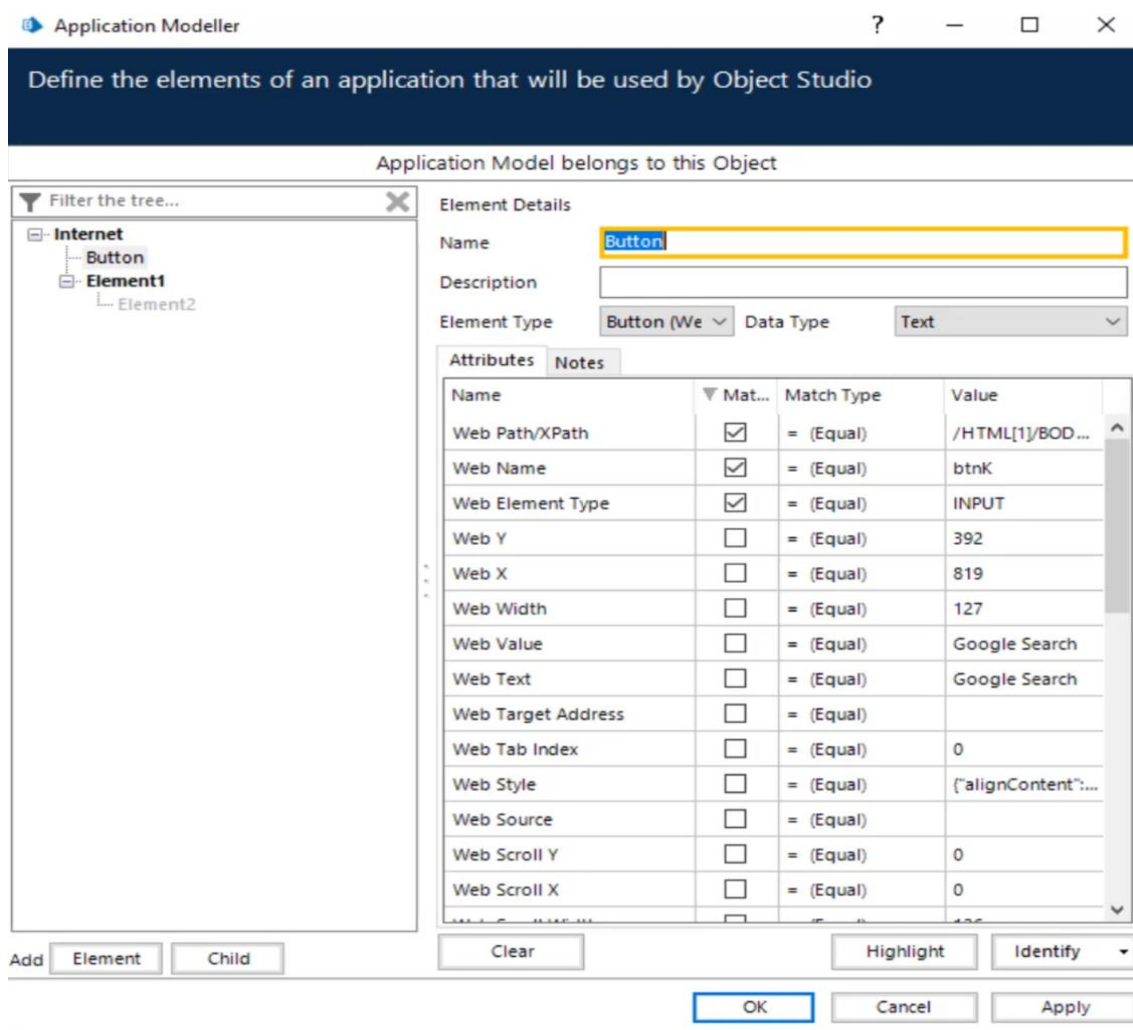
Blue Prism -mallinnus koostuu kahdesta alueesta, prosessipuolesta (Process Studio) ja objektipuolesta (Object Studio). Prosessipuolelle rakennetaan itse robotin logiikka, ja se usein sisältää useita eri alisivuja. Objektipuolella luodaan robotin käyttämät toiminnot, kuten painikkeen klikkaus tai taulukon lukeminen. Kuvassa 6 esitetään esimerkkimalli robotin prosessista Blue Prismin Process Studion puolelta. Kuvasta näkee, että Blue Prismin logiikka muistuttaa hyvin paljon perinteisiä prosessikaavioita.



Kuva 6. Blue Prism -robotin prosessimalli

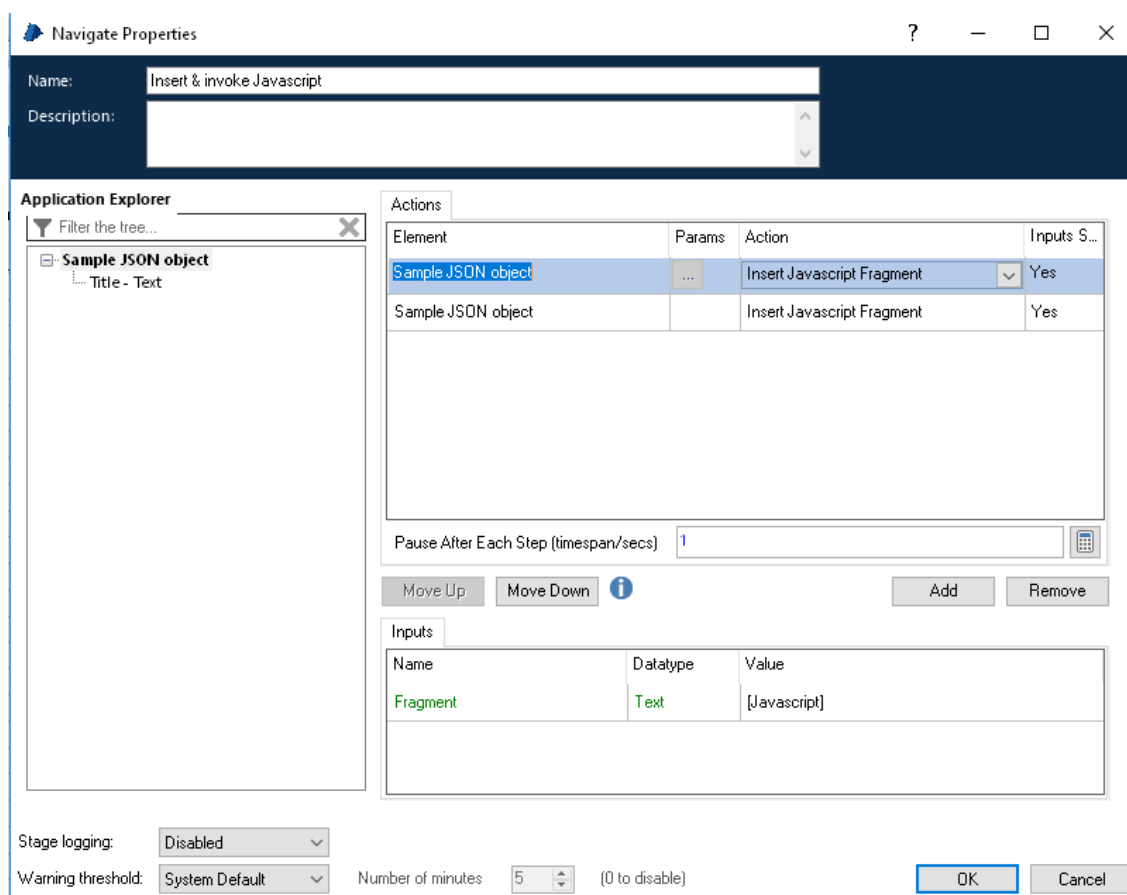
Objektipuolella on myös Blue Prismin yksi tärkeimmistä työkaluista: Application Modeller, jota käytetään käyttöliittymän elementtien tunnistukseen. Selaimesta

tunnistamiseen Blue Prism tarvitsee selaimen liitetyn Browser Extension -lisäosan asennettuna. Application Modeller pystyy tunnistamaan tuettujen selainten lisäksi Java-, Windows- ja Mainframe -sovelluksia. Java -sovellusten tunnistamisessa hyödynnetään Java Access Bridge -työkalua, jonka avulla Blue Prism pääsee käsiksi sovelluksen käyttöliittymän lähdekoodiin. Tämä työkalu tulee siis olla asennettuna Blue Prism -resurssikoneella. Kuvassa 7 esitetään elementin tunnistus Googlen hakusivulta käyttäen Chrome-selainta. Application Modeller tukee selainpohjaisessa tunnistuksessa myös XPath -syntaksia, jolloin hankalammissa tilanteissa voidaan luoda itse tunnistussäännöt. Esimerkiksi tilanteessa, jossa sivulla on useita samankaltaisia elementtejä samoilla tunnistustiedoilla, eikä haluta luottaa tunnistuksessa puhtaasti indeksitietoon. Indeksitieto tarkoittaa tässä yhteydessä löydettyjen tulosten lukumäärää eli sitä, monesko tulos halutaan valita. Tällaisissa tilanteissa voidaan luoda sopiva tunnistus XPath-syntaksia hyödyntäen. Se on kyselykieli XML-pohjaisten dokumenttien ositukseen, jonka avulla voidaan tunnistaa elementtejä HTML-lähdekoodista [12]. Ohessa on esimerkki XPath-syntaksista, jossa tunnistetaan HTML-lähdekoodista painike. Painikkeen ID on 'testi' ja sen edeltävä sisärelementti on DIV, joka sisältää tekstin 'Testiotsikko'. Esimerkki: `//button[@id='testi' and preceding-sibling::div[@text()='Testiotsikko']]`.



Kuva 7. Application Modeller

Application Modeller ei kuitenkaan aina tunnista edes selainpohjaisesta käyttöliittymästä kaikkia elementtejä. Esimerkiksi upotetut elementit, kuten IFrame ja Shadow Root tuottavat Application Modellerille usein hankaluuksia. Mikäli kehittäjällä on ohjelmointikielistä osaamista, niin yllämainittujen ongelmaelementtien manipulointiin Blue Prism tarjoaa mahdollisuuden lähettää JavaScript-funktioita suoraan halutulle sivulle, kuten kuvassa 8 esitetään. Tämän toiminnon avulla kehittäjä pystyy tunnistamaan huomattavasti laajemmalla skaalalla elementtejä ja manipuloida näitä Blue Prismille tunnistettavaksi omien kykyjensä rajoissa.



Kuva 8. JavaScriptin hyödyntäminen Blue Prismissä [13]

Blue Prism sisältää myös valmiin login-agent -toiminnon, jolla robotit pystyvät kirjautumaan itse suoraan resurssikoneelle ja samaan tapaan kirjautumaan ulos, kun työt on suoritettu loppuun.

Kaikkea tätä ei kuitenkaan saa ilmaiseksi. Blue Prism on lisenssihinnoiteltu saman aikaisesti ajossa olevien robottien lukumäärän perusteella.

2.3.2 Python

Python on suosittu ja monipuolinen ohjelmointikieli, joka on yksi helpoiten lähes-tyttävistä ohjelmointikielistä sen yksinkertaisen syntaksin ansiosta. Pythonin historia juontaa juurensa 1980-luvun loppupuolelle, kun se tehtiin alun perin jatkoksi ABC-ohjelmointikielelle. Python 3.0 -versio julkaistiin joulukuussa 2008 ja

tällä hetkellä uusin vakaa versio on 3.10. Pythonia käytetään automaation ja ohjelmistorobotiikan ohella mm. data-analytiikassa, koneoppimisessa, datalouhinnassa, testiautomaatiossa ja tekoälysovelluksissa. [14.]

Python soveltuu automaation ja ohjelmistorobotiikan tekoon sen monipuolisuuden, helppouden ja laajan kirjastotarjonnan takia. Yksi tärkein kirjasto automaation ja ohjelmistorobotiikan kannalta on jo vuosia ollut web-pohjaiseen testiautomaatioon luotu Selenium-kirjasto, jonka avulla pystyy luomaan käyttöliittymämallinnusta samalla tavalla kuin valmiilla RPA-työkaluilla. [14.]

Selenium on toiminut käyttöliittymätestauksen ja automaation kulmakivenä jo vuosikaudet, mutta tämän käytössä on myös ollut omat ongelmansa. Yksi suurimmista ongelmista on, että Selenium tarvitsee erikseen tietokoneelle asennettuna selainajurin, jotta se pääsee selaimen käsiksi. Nykyisin selaimet päivittyvät jatkuvalla tahdilla, joten jos haluaa pitää selaimen päivitettyinä, niin ajuri tarvitsee päivittää samalla. Tämän ongelman ratkaisee Microsoftin luoma uusi tuotos PlayWright-kirjasto, joka on alun perin luotu Googlen luomasta Puppeteer-kirjastosta. PlayWright-paketti sisältää itsessään selainkomponentit, jolloin käyttäjän ei tarvitse päivittää ajureita manuaalisesti. Tätä toiminnallisuutta hyödynnetään myös Robot Frameworkin tekijöiden luomassa uudessa Browser-libraryssä, mikä käyttää PlayWrightiä taustalla. [15; 16.]

Python on oiva valinta ohjelmistorobotin tekoon, jos robotti halutaan tehdä puhtaasti ohjelmointikielellä ja jos automatisoitava prosessi tarvitsee toiminnallisuksia, joita ei löydy muista työkaluista. Seuraavaksi esiteltävä Robot Framework on myös hyvä ottaa huomioon, sillä se on Pythoniin pohjautuva sovel-luskehys. Siihen pystyy suoraan yhdistämään itsetehtyjä Python-kirjastoja, mutta se myös sisältää paljon valmista toiminnallisuutta automaatiota ja robo-tiikkaa ajatellen. Kummassakaan teknologiassa ei ole suoraan valmista työka-lua elementtien tunnistukseen, vaan järjestelmän taustakoodi pitää itse käydä tutkimassa ja luoda tarvittavat tunnistukset. Selainpohjaisissa järjestelmissä voi-daan hyödyntää esimerkiksi XPath-syntaksia. Python ja Robot Framework ovat molemmat ilmaisia teknologioita ohjelmistorobotin kehitykseen.

2.3.3 Robot Framework

Robot Framework on Suomessa luotu avoimen lähdekoodin sovelluskehys. Se sai alkunsa Pekka Klärckin diplomityöstä Nokia Networksille vuonna 2004, mutta sen ensijulkaisu tapahtui vuonna 2008. Robot Framework kehitettiin alun perin testiautomaation tehostamiseen, mutta nykyisin se on laajentunut ohjelmistorobotiikkaan. Sillä on laaja intohimoinen kehittäjäyhteisö taustalla, varsinkin Suomessa. [17.]

```

*** Settings ***
Documentation      Reusable keywords and variables.
Library           SeleniumLibrary

*** Variables ***
${SERVER}         localhost:7272
${BROWSER}       Firefox
${LOGIN URL}     http://${SERVER}/
${WELCOME URL}   http://${SERVER}/welcome.html

*** Keywords ***

Open Login Page
    Open Browser    ${LOGIN URL}    ${BROWSER}
    Login Page Should Be Open

Login Page Should Be Open
    Location Should Be    ${LOGIN URL}
    Title Should Be      Login Page

Input Credentials
    [Arguments]    ${username}    ${password}
    Input Text     username_field  ${username}
    Input Text     password_field  ${password}

Submit Credentials
    Click Button   login_button

Welcome Page Should Be Open
    Location Should Be    ${WELCOME URL}
    Title Should Be      Welcome Page

```

Kuva 9. Esimerkki Robot Framework -syntaksista. [18]

Robot Framework -kehitys tapahtuu ohjelmointiympäristössä, mutta sen koodi on hyvin helppoa luettavaa verrattuna perinteisiin sovelluskehyyksiin tai ohjelmointikieliin. Kuvassa 9 on esimerkki Robot Framework -syntaksista, jossa esitetään yksinkertainen kirjautuminen selainsovellukseen.

Robot Framework sisältää useita eri kirjastoja, jotka on luotu testiautomaatiota ja ohjelmistorobotiikkaa varten. Se sisälsi vielä 3.2-versioissa jonkin puutteita ja rajoituksia ja siitä huomasi, että se on selvästi testiautomaatiota varten kehitetty. Uudessa 4.0-versiossa on otettu selvä harppaus kohti ohjelmistorobotiikkaa ja puutteita on korjattu. Nykyisin FOR-loopit voivat olla toistensa sisällä, natiiivi IF-ELSE-syntaksi on tuettu ja "test case" voi olla nykyisin myös "task". [19.]

Robot Frameworkin yksi tärkein ominaisuus on sen skaalautuvuus myös muihin kieliin. Siihen on mahdollista liittää mukautettuja Python- tai Java-kirjastoja, mikäli valmiit kirjastot tai toiminnallisuudet eivät riitä. Liittäminen tehdään avainsanoja (keywords) hyödyntämällä, jolloin Robot Framework -toteutuksista voidaan tehdä lähes yhtä kattavia kuin esimerkiksi puhtaasti Python-toteutuksista. [20.]

COMPARISON BETWEEN OPEN AND CLOSED RPA SOLUTIONS

	Open solution	Closed solution
Licence costs	None	Yes, depending on orchestration and number of parallel processes
Scalability	Yes with open source tools	Yes with additional cost
Features	Limitless extensibility	Vendor supported feature set
Security	Open for audit	Closed, based on trust
Support for using machine learning models	Yes, state of the art tools available	Vendor supported feature set
Resources for developers	Active global community of thousands of developers and contributors, free online resources and documentation	Vendor specific programs

Kuva 10. Avoin vastaan suljettu ympäristö. [20]

Robot Framework on avoimen lähdekoodin sovelluskehys, ja tämä on sen vahvuus verrattuna valmiisiin ohjelmistorobottisovelluksiin. Kuvassa 10 näkyy vertailua avoimen ja suljetun ratkaisun välillä. Avoimen ratkaisun vahvuuksia on sen läpinäkyvyys auditoinnille, yhteisön tuki ja laajennettavuus. Avoin ratkaisu

on myös luontaisesti ilmainen, mikä on yksi avaintekijä Robot Frameworkin suosion taustalla.

2.4 Hyödyt

Ohjelmistorobotiikan yksi merkittävin hyöty on toistuvien tehtävien automatisointi eli manuaalityön vähentäminen. Tämän ansiosta voidaan vapauttaa työntekijöiden aikaa yritykselle arvokkaampiin ja hyödyllisempiin tehtäviin ja samalla nostaa asiakastyytyväisyyttä ja pienentää virhemarginaaleja. Robotti pystyy työskentelemään kellon ympäri viikon jokaisena päivänä ilman uupumusta tai hidastumista ja usein moninkertaisesti nopeammin kuin ihminen. Tämä tarkoittaa sitä, että asiakkaat saavat nopeammin asiansa käsittelyyn, jolloin yrityksen tehokkuus ja asiakastyytyväisyys myös nousee. Leadership Finlandin artikkelissa haastateltiin ohjelmistorobotiikkaa hyödyntävän Eera Oy:n toimitusjohtajaa Erik Flomania. Hän mainitsi asiasta seuraavasti: ”RPA luo myös uusia liiketoimintamahdollisuuksia teknologioista ja palveluista. Se on myös ratkaisu työvoimavajeeseen, mikä syntyy väestön ikääntyessä.” Väittämässä on perää, sillä ohjelmistorobotiikan markkina-arvon odotetaan monikertaistuvan seuraavan kymmenen vuoden aikana. [21; 22.]

Ohjelmistorobotin hyöty on myös sen helppous verrattuna järjestelmäautomaatioihin. Robotti ei tarvitse erillisiä integraatioita eikä käytössä olevaan järjestelmään tarvitse tilata muutoksia. Käyttöönotto on myös usein pienempi kokonaisprosessi, sillä ohjelmistorobotin tekeminen on huomattavasti nopeampaa ja vaatii pienemmän työmäärän kuin järjestelmään rakennettava automaatio. Käyttöönotossa puhutaan usein viikoista kuukausien sijaan. Tällä on myös suora vaikutus kustannuksiin, ja robotti on yksinkertaisissa liiketoimintaprosesseissa usein se halvempi vaihtoehto.

2.5 Rajoitteet

Ohjelmistorobotti käyttää usein sovelluksia käyttöliittymän kautta samalla tavalla kuten ihminen näitä käyttäisi eli toimii front-end-puolella. Tämä eroaa perin-

teisestä automaatiosta, sillä nämä toimivat järjestelmien taustapuolella (back-end) hyödyntäen sovelluksiin luotuja rajapintoja ja valmiita integraatioita. Tämä mahdollistaa automaation hyödyntämisen järjestelmiin, missä sitä ei muuten voi käyttää, mutta toimii samalla yhtenä ohjelmistorobotiikan suurimmista rajoitteista. [23.]

Käyttöliittymämallinnus nojaa pitkälti maltillisten järjestelmäpäivitysten varassa, sillä suuremmat front-end-koodimuutokset rikkovat helposti robotit. Tähän yksi ratkaisu on luoda mahdollisimman skaalautuvat ja yleiskäyttöiset käyttöliittymämallinnukset, joissa tunnistetaan elementin attribuuteista vain osa hyödyntäen "wildcard"-tunnistuksia. Tämänkaltaisessa tunnistuksessa on kuitenkin riskinsä, sillä mallinnus saattaa mahdollisen käyttöliittymämuutoksen yhteydessä tunnistaa väärän elementin ja toimia alkuperäisten määrittelyiden vastaisesti.

Ohjelmistorobotti toimii usein yrityksillä hyvänä tilapäisenä ratkaisuna, kun isompi automaatio ei ole mahdollista tai halutaan viivyttää suurempaa investointia uuteen järjestelmään tai integraatioiden luontiin. Tässäkin on kuitenkin vaaransa, sillä ohjelmistorobotteja hyödyntämällä yritys saattaa liikaa jarruttaa uusien järjestelmien hankintaa. Tällöin lopputulos on helposti vanhentunut ja huonosti palveleva järjestelmä, johon on rakennettu useita robotteja paikkaamaan sen puutteita. Tämä johtaa helposti tilanteeseen, jossa toimiva kokonaishallinta on lähes mahdotonta.

Ihmiset saattavat suhtautua robotiikkaan varauksella, ja vastaanotto saattaa olla jopa hyvinkin negatiivista. Työntekijöiden negatiivinen suhtautuminen aiheuttaa helposti varsinkin isommissa organisaatioissa sen, että kaikkia automatisoivissa olevia prosesseja ei välttämättä tunnisteta, sillä ajatellaan, ettei robotti tätä kuitenkaan pysty tekemään tai se on liian tärkeä robotille annettavaksi. Varsinkin käyttöönoton alussa työntekijöiden suhtautuminen robotiikkaan saattaa olla negatiivista, mutta tätä voi helpottaa yrityksen sisäisillä koulutuksella ja tarpeeksi hyvällä tiedonkululla. [23.]

Perinteisessä ohjelmistorobotiikassa yksi rajoite on myös sen perustoiminnallisuus, eli ennalta määritellyt säännöt ja näiden mukaan toiminen. Tämä aiheuttaa rajoituksia robotin toimintaan poikkeavissa tilanteissa, sillä pieninkin sääntöjen ulkopuolinen tilanne siirretään manuaalikäsittelyyn, ja tällöin automaatioaste pienenee.

3 Automatisoitava prosessi

Kaikki liiketoimintaprosessit eivät ole automaattisesti sopivia ohjelmistorobotille, ja näiden tunnistaminen on tärkeää robotin esiselvitysvaiheessa. Vaikka ohjelmistorobotit ovat yleensä kevyemmän toteutuksen takana kuin perinteinen automaatio, niin se on silti kallis tapa automatisoida pieniä prosesseja, joissa robotin tuomia etuja ei päästä hyödyntämään.

3.1 Liiketoimintaprosessi

Millainen on hyvä liiketoimintaprosessi ohjelmistorobotille? Keskeisin asia robotille sopivassa liiketoimintaprosessissa on se, että robotti pystyy tuottamaan siihen käytetyt resurssit takaisin eli luomaan arvoa. Prosessit, jotka vaativat työntekijöiltä paljon manuaalityötä, tarkkaavaisuutta ja toistoja ovat täydellisiä roboteille, sillä näissä saadaan useimmiten eniten hyötyä ohjelmistorobotin käytöstä. Tässä hyvänä esimerkkinä toimii esimerkiksi valtavat kertajaot, jossa puhutaan tuhansien rivien kirjaamisesta useampaan eri järjestelmään. [24; 25.]

Liiketoimintaprosessin sopivuutta ohjelmistorobotille voidaan tarkastella usealla eri tavalla, joista yksi on selvittää, miten paljon prosessi vie käsittelijöiltä aikaa vuodessa. Kuvassa 11 esitetään esimerkkiprosessin laskukaava, missä laskeetaan työntekijän käyttämä aika tapauksien kirjaamiseen järjestelmiin. Kaavassa "lkm" tarkoittaa tapausten määrää vuodessa ja "aika" yhden tapauksen läpivientiin menevä aika minuutteina. Esimerkiksi tapauksia voisi olla yhteensä vuodessa 10 000 kappaletta ja yhden tapauksen tekemiseen menee työntekijältä keskimäärin 4 minuuttia. Jaetaan näiden tulo tunneiksi ja tämä jaetaan vielä työpäiviksi käyttäen 7,5 tunnin pituisia työpäiviä. Esimerkkitapauksessa lopputulokseksi saadaan pyöristettynä 89 työpäivää. Mitä suurempi tämä luku on, sitä enemmän ohjelmistorobotti tuo hyötyä prosessin automatisoinnissa.

$$\frac{\left(\frac{lkm * aika}{60}\right)}{7,5}$$

Kuva 11. Esimerkkikaava hyödyn laskemisen tueksi

Ennen ohjelmistorobotin hyödyntämistä tulee myös ottaa huomioon järjestelmät, mitä käytetään prosessissa. Rajapintojen puuttuminen ei haittaa ohjelmistorobottia, mutta koodissa olevat poikkeamat saattavat aiheuttaa ongelmia ja nämä on hyvä selvittää ennen kehitystä. Näitä ovat esimerkiksi IFramet ja Shadow Rootit selainpohjaisissa käyttöliittymissä tai RDP-etäyhteyden takana olevat sovellukset. Näin ollen robotin esiselvitysvaiheessa tulee tutkia käytettävien järjestelmien koodia ja sitä, miten hyvin käytössä olevalla robottiteknologialla saadaan käyttöliittymässä tunnistettua elementtejä.

Kaikki järjestelmät eivät salli ohjelmistorobottien käyttämistä, vaikka se teknologian puolesta olisi mahdollista. Tämä on siis tärkeä tarkastaa ennen robotin kehitystä, sillä mikäli tämä käy ilmi jälkikäteen, seuraukset voivat olla ikävät. Tämän kuitenkin saa usein selville suoraan järjestelmän tietoja tarkastellessa. Tämänkaltaiset sovellukset usein kuitenkin tarjoavat maksullista rajapintaa suurempien massojen käsittelyyn. Hyvä esimerkki vastaavasta sovelluksesta on Yritys ja Yhteisötietojärjestelmä YTJ, missä yrityshaun käyttäminen hakuroboteilla on kielletty. YTJ tarjoaa kuitenkin 1240 euron avausmaksulla rajapintaa yrityksille käyttöön. [26.]

3.2 Datan laatu

Datan laatu on erittäin isossa roolissa missä tahansa liiketoiminnassa ja varsinkin automaatioon liittyvissä tehtävissä. Laadukas data auttaa yrityksiä toimimaan tehokkaammin ja tuo kilpailuetua markkinoilla. [27.]

Ohjelmistorobotti voi olla osa yrityksen datan laadunhallintaa, sillä se ei tee näppäilyvirheitä tai muita inhimillisiä virheitä, mitä ihmisille voi tapahtua. Se voi myös toimia osana suurempaa automaatiototeutusta, esimerkiksi validoimalla datan laatua ja rikastaa tätä. Datan laadunhallintaan panostaminen tehostaa prosesseja, vähentää virheiden esiintymistä, parantaa asiakaskokemusta, helpottaa tiedon löytämistä ja mahdollistaa suurempaa analytiikkaa ja automaatiota. [27.]

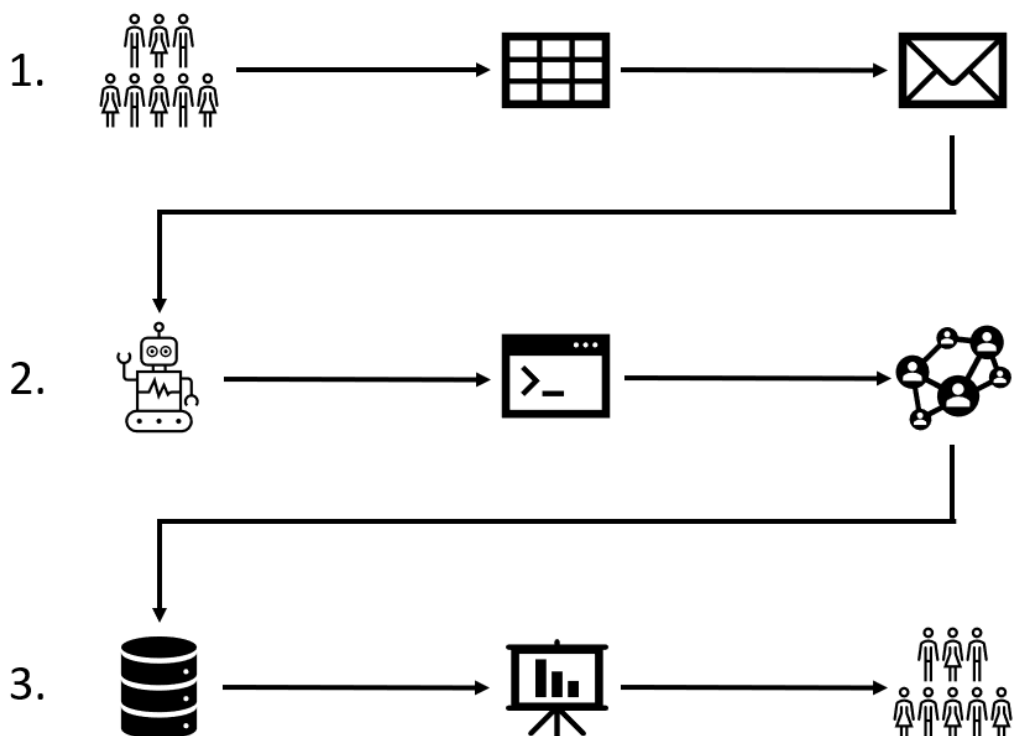
Datan laatu vaikuttaa kuitenkin myös ohjelmistorobottien toimivuuteen. Mikäli ohjelmistoroboteille saapuva data ei vastaa määriteltyä muotoa, niin robotin onnistumisprosessi laskee huomattavasti, sillä tapausta ei pystytä tekemään suoraan läpi ennen kuin data on korjattu. Tämän takia robotille saapuva lähtödata on optimaalisimmillaan, kun se saapuu validoituna ja määrämuotoisena toisesta järjestelmästä, esimerkiksi kenttävalidoidusta lomakkeesta.

4 Esimerkkirobotti

Tässä luvussa kuvataan esimerkkirobotin toiminta. Robotti pohjautuu todelliseen toteutukseen, mutta kaikki tunnistettavissa olevat yksityiskohdat on jätetty pois.

4.1 Kokonaisprosessi

Prosessi on jaettu kolmeen osaan ja tämä esitetään kuvassa 12. Ensimmäinen osa kuvaa myynnin roolia prosessissa. Myyjät työskentelevät asiakasrajapinnassa ja tekevät kauppvoja niin uusille kuin vanhoille asiakkaille. Myyntitapahtumat kirjataan Excel-tiedostoon ja tiedosto lähetetään päivittäin ohjelmistorobotin sähköpostilaatikkoon. Tiedosto sisältää listauksen kaikista päivän myyntitapahtumista, eli myydyt tuotteet ja asiakkaiden ajankohtaiset asiakastiedot, jotka myyjät ovat selvittäneet asiakkailta myyntitapahtumien yhteydessä.



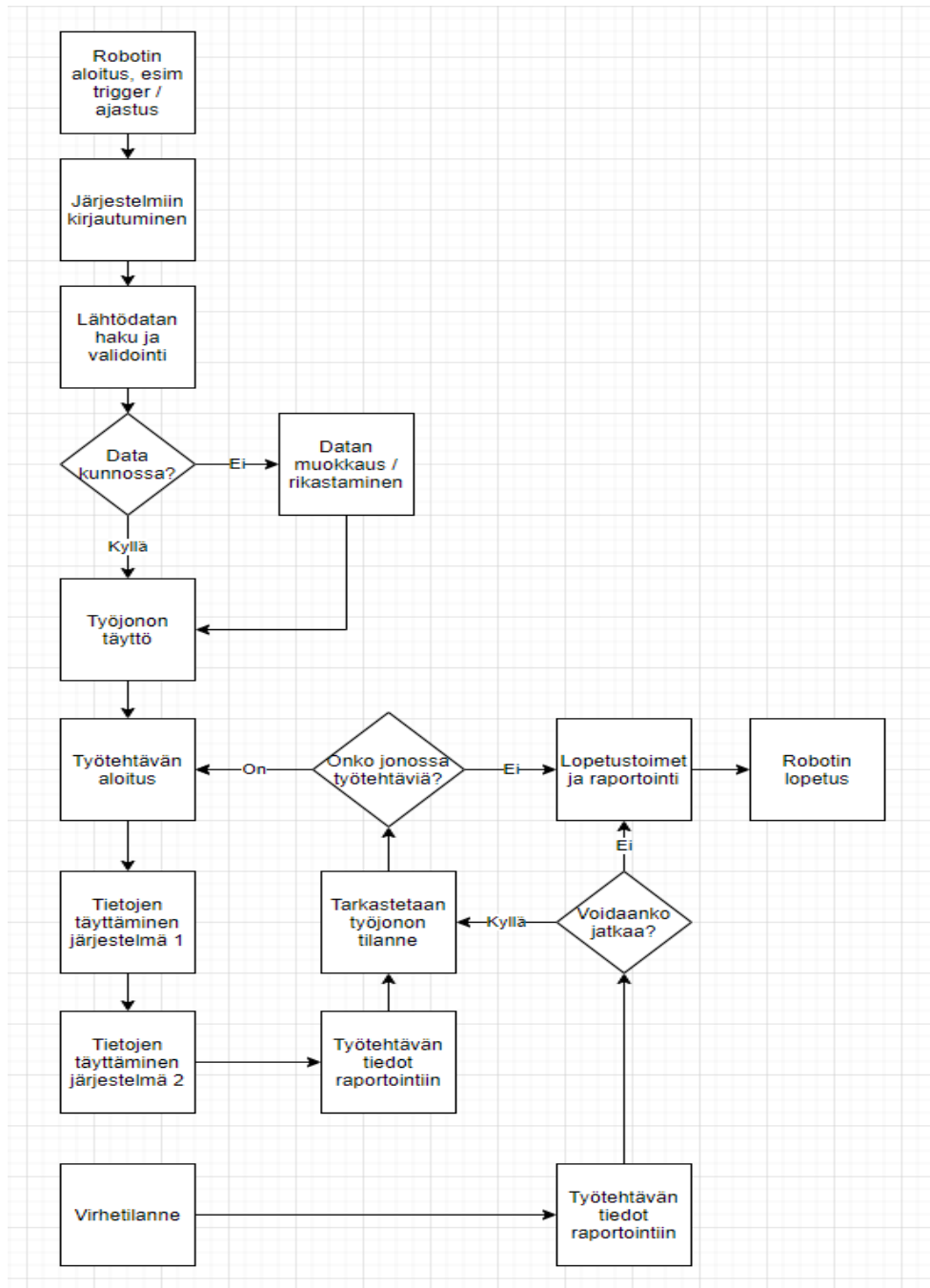
Kuva 12. Esimerkkirobotin kokonaisprosessi

Toinen osa sisältää ohjelmistorobotin työosuuden. Ohjelmistorobotti ajaa kerran päivässä ja hakee sille luodusta sähköpostilaatikosta kaikki viestit, jotka noudattavat prosessissa sovittuja kriteerejä ja sisältävät robotille määritellyt avainsanat. Robotti kirjautuu yrityksen myyntijärjestelmään ja kirjaa tänne listan myyntitapahtumat. Seuraavaksi robotti kirjautuu yrityksen asiakashallintajärjestelmään ja päivittää tai lisää asiakkaiden tiedot tänne.

Kolmannessa osassa kuvataan prosessin raportointia. Jokaisen kirjatun tapahtuman jälkeen robotti vie tapauksen tiedot tietokantaan. Tietokannasta luodaan liiketoiminnalle PowerBi-näkymä, josta yrityksen johto voi tarkastella päivittäisiä myyntitapahtumia ja robotin tekemiä kirjauksia.

4.2 Robotin prosessi

Ohjelmistorobotin prosessikaavio esitetään kuvassa 13, josta näkee prosessin kulun robotin näkökannalta. Robotti käynnistetään sille luodulla ajastuksella. Tässä voidaan myös käyttää impulssia eli triggeriä käynnistyksessä. Ajastukseen voi käyttää useita eri työkaluja, mutta suosituimmissa RPA-työkaluissa on usein valmiit toiminnallisuudet ajastuksien hallintaan, esimerkiksi Blue Prism Scheduler ja UiPath Orchestrator. Python- tai Robot Framework -toteutuksissa voidaan hyödyntää esimerkiksi Jenkins-työkalua ajastushallinnassa. Myös pilvipalveluilla on usein omat sisäänrakennetut ajastustoiminnallisuudet, mikäli robottiympäristö on rakennettu pilveen. Windows-ympäristöissä voidaan hyödyntää myös Windowsin omaa Task Scheduleria. Esimerkkirobotin tilanteessa robotti kirjautuu resurssikoneelle käyttäen Blue Prism -sovelluksen omaa LoginAgent-työkalua. Resurssikoneen käyttö ja sinne kirjautuminen vaihtelee riippuen käytössä olevasta ohjelmistorobottiympäristöstä. [28.]



Kuva 13. Ohjelmistorobotin prosessikaavio

Käynnistyksen jälkeen robotti käy kirjautumassa käytettäviin sovelluksiin. Tämä on tärkeää tehdä prosessin alkuvaiheessa, sillä tässä testataan samalla järjestelmien toimivuus. Mikäli tämä tehtäisiin myöhemmässä vaiheessa prosessia,

niin robotti menisi turhaan virheeseen keskellä prosessia ja työtehtävä jäisi turhaan virheeseen, jolloin tämä jouduttaisiin todennäköisesti ohjaamaan liiketoiminnalle manuaalikäsittelyyn.

Riippuen automatisoitavasta prosessista lähtödatan haku voi tapahtua lähes mistä vaan, kuten sähköpostista, tietokannasta, pilvipalvelimelta tai esimerkiksi suoraan tietokoneen levyasemalta. Esimerkkirobotti hakee lähtödatan sen omasta Outlook-sähköpostilaatikosta ja tallentaa Excel-tiedoston väliaikaisesti resurssikoneen lokaaliin kansioon. Lähtödatan oikeellisuus validoidaan tässä vaiheessa ja poistetaan esimerkiksi duplikaatit datasta. Lähtödatan haun jälkeen robotti kirjaa datan työjonolle. Esimerkkitapauksessa hyödynnetään Blue Prismin omaa työjonohallintatyökalua.

Robotti ottaa työjonosta tapauksen käsittelyyn ja tallentaa muuttujiin tapauksen tiedot. Esimerkkirobotin prosessissa on kaksi järjestelmää, mihin tiedot täytetään yksitellen. Molemmissa järjestelmissä on ennalta määritelty raja-arvo uudelleen yrityksille, mikäli prosessissa tulee odottamaton virhe vastaan. Näitä ovat esimerkiksi järjestelmän yhteyskatkos tai muu vastaava hetkellinen häiriö. Työosuuden jälkeen tapauksen tiedot tallennetaan raportointimuuttujiin ja tarkastetaan, onko työjonolla lisää tapauksia. Mikäli robotilla ei ole valmista työnokäsittelyä, niin ulkoiseen tietokantaan tallennus on järkevää, jotta tietoja ei menetä virhetilanteessa.

Lopetustoimissa on järjestelmistä uloskirjautuminen, väliaikaisen datan poistaminen tai arkistointi ja ajon raportointi. Raportointi voidaan hoitaa myös tapauskohtaisesti jokaisen käsitellyn tapauksen jälkeen. Raportoinnin voi toteuttaa usealla eri tavalla, esimerkiksi SQL-kantaa hyödyntäen, josta saadaan suoraan PowerBi-näkymä tai se voidaan hoitaa suoraan sähköpostilla.

Toimivan ohjelmistorobotin yksi tärkein osa on virhekäsittely. Blue Prism -toteutuksissa virheet jaetaan kolmeen eri kategoriaan. Näitä ovat System Exception eli järjestelmävirhe, Business Exception eli sääntövirhe ja Internal eli sisäinen virhe. Mikäli robotti kohtaa prosessissa järjestelmävirheen, yritetään tapausta

uudelleen tiettyyn raja-arvoon asti tai kirjataan tapauksen tiedot manuaalikäsittelyyn ja jatketaan seuraavasta tapauksesta. Sama koskee sisäisiä virheitä. Mikäli sama odottamaton virhe esiintyy useamman kerran peräkkäin, voidaan käsittely lopettaa. Sääntövirheet ovat ennalta määriteltyjä virheitä, jotka siirretään suoraan manuaalikäsittelyyn, eikä näitä turhaan yritetä uudelleen. Sääntövirheiden raja-arvot tulevat usein suoraan liiketoiminnalta. [29.]

5 Ohjelmistorobotiikka tulevaisuudessa

Ohjelmistorobotiikan ennustetaan kasvattavan suosiotaan seuraavien vuosien aikana ja siitä tulee tällöin yhä yleisempää myös pienemmissä yrityksissä. Perinteisen ohjelmistorobotiikan rajoitteena on kuitenkin, että se pohjautuu pelkästään ennalta määriteltyihin sääntöihin. Suoraan asiakkailta tuleva vapaamuotoinen data tuottaa perinteisille roboteille haasteita, ja näiden ratkaisu on usein roboteille uusien sääntöjen opettaminen. Tähän on kuitenkin toinenkin ratkaisu, ja se on laajentaa ohjelmistorobotteja tekoälyllä ja koneoppimisella.

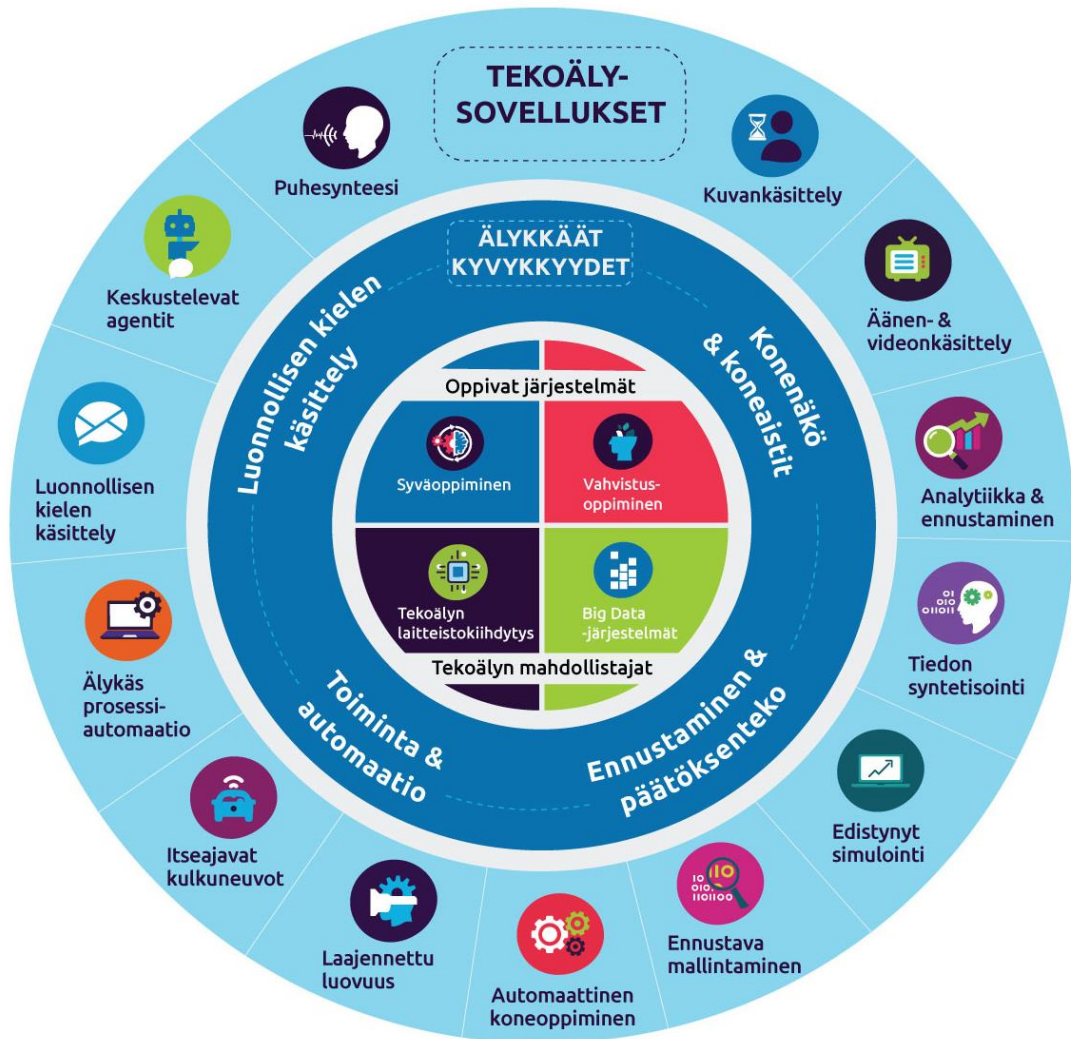
5.1 Tekoäly ja koneoppiminen

Tekoälyllä tarkoitetaan koneen kykyä käyttää ihmismäistä päättelyä. Tekoälystä käytetään myös usein termiä AI eli Artificial Intelligence. Termiä käytetään paljon koneoppimiseen perustuvista analytiikkaratkaisuista, joiden avulla pystytään automatisoimaan päätöksentekoa. Machine Learning (ML) eli koneoppiminen on tekoälyn keskeinen osa-alue, jonka tarkoituksena on päätyä haluttuun lopputulokseen ilman valmiiksi määriteltyä algoritmia tai toimintamenetelmää. [30.]

Tekoäly jaetaan kahteen luokkaan: heikkoon ja vahvaan tekoälyyn. Heikolla tekoälyllä tarkoitetaan sovellusta, jolla ei ole tietoisuutta ja kykenee toimimaan yhdellä osa-alueella. Kaikki nykyiset tekoälytoteutukset perustuvat heikkoon tekoälyyn. Vahva tekoäly taas tarkoittaa kykyä kehittää tietoisuus. [30.]

Kuvassa 14 esitetään tekoälyn osa-alueita ja miten tekoälyä voidaan hyödyntää erilaisissa sovelluksissa. Näitä ovat muun muassa kuvankäsittely, analytiikka, kielen käsittely, ennustava mallintaminen ja älykäs prosessiautomaatio. Näistä useat ovat toiminnallisuuksia, jotka auttavat ohjelmistorobotiikkaa kehittymään parempaan suuntaan.

Tekoälyn osa-alueita



Kuva 14. Tekoälyn osa-alueita [31]

Tekoälyratkaisuja kehitetään jatkuvasti lisää ja niitä hyödynnetään myös ohjelmistorobotiikassa. Valmiit ohjelmistorobotiikkasovellukset ovat jo tuoneet versio-päivityksissä tekoälyä hyödyntäviä ominaisuuksia ja tuovat jatkuvasti lisää koneoppimis- ja tekoälytyökaluja hyödynnettäväksi. Tulevaisuudessa tekoälyn hyödyntäminen ohjelmistorobotiikassa vain kasvaa.

5.2 Hyperautomaatio

Hyperautomaatio on lähivuosina ollut paljon pinnalla tulevaisuuden ratkaisuna. Hyperautomaatio on useamman eri teknologian yhdistelmä, missä hyödynnetään yhtä tai useampaa ohjelmistorobottiteknologiaa, järjestelmäautomaatiota ja tekoälyä. Ohessa listattuna hyperautomaatiossa tyypillisesti esiintyviä ominaisuuksia ja teknologioita. [32.]

- ohjelmistorobotiikka
- koneoppiminen
- prosessilouhinta
- tehtävälouhinta
- tekstintunnistus
- tekoälyratkaisut
- low-code- ja no-code -teknologiat
- luonnollisen kielen käsittely
- keskustelevat käyttöliittymät
- analytiikka.

Listatuista ominaisuuksista yhä useampi on ilmaantunut uusimpien versiopäivityksien myötä valmiisiin ohjelmistorobotiikkaratkaisuihin, jolloin hyperautomaation hyödyntäminen perinteisessä ohjelmistorobotiikassa tulee kasvamaan. Esimerkiksi Blue Prism markkinoi jatkuvasti yhä enemmän termillä ”Intelligent Automation Platform”, jonka uudet työkalut sisältävät useita hyperautomaation ominaisuuksia. Uusimmissa ohjelmistorobotiikkateknologioiden versioissa on

myös paranneltu hankalien HTML-elementtien tunnistamista. Blue Prism ja Robot Framework ovat molemmat uusissa versioissa tuoneet CSS selector -teknologiaan pohjautuvaa tunnistusta, jonka avulla pystyy helpommin tunnistamaan myös upotettuja elementtejä, kuten IFrame ja Shadow Root. [15; 33.]

6 Yhteenveto

Työn tarkoituksena oli tutkia ohjelmistorobotin roolia liiketoimintaprosessin automatisoinnissa, robotin kehitysprosessia ja vertailla muutamaa olemassa olevaa työkalua. Tämän lisäksi haluttiin tarkastella ohjelmistorobotiikan mahdollisia ongelmia ja miettiä robotiikan tulevaisuutta liiketoimintaprosessien automatisoinnissa.

Työn sisältö ja tarkoitus hieman muuttuivat kesken prosessin työnantajavaihdoksen myötä, mutta työssä saatiin silti esitettyä kattava kuva ohjelmistorobotiikasta ja siitä, miten sitä voidaan hyödyntää liiketoimintaprosessien automatisoinnissa. Työssä vertailtiin ohjelmistorobotiikan keskeisimpiä teknologioita ja työtä voi käyttää teknologian valinnan tukena.

Ohjelmistorobotiikka on tehokas tapa automatisoida manuaalityötä vaativia liiketoimintaprosesseja, ja sen hyödyntäminen tulee tulevaisuudessa kasvamaan entisestään.

Lähteet

- 1 Staria. Mitä on ohjelmistorobotiikka? Verkkoaineisto. <https://staria.com/fi/blogi/mita-ohjelmistorobotiikka/>. Luettu 25.3.2021.
- 2 Srinsoft. RPA as a service (RPAaaS). Verkkoaineisto. <https://www.srinsoft-tech.com/rpa-as-a-service-rpaas.html>. Luettu 25.3.2021
- 3 Digital Workforce. Ohjelmistorobotiikka. Verkkoaineisto. <https://digitalworkforce.com/fi/digityontekija/rpa-ohjelmistorobotiikka/>. Luettu 25.3.2021
- 4 Valtioneuvoston selvitys -ja tutkimustoiminta. Ohjelmistorobotiikka ja tekoäly -soveltamisen askelmerkkejä. Verkkoaineisto. <https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161123/65-2018-Ohjelmistorobotiikka%20ja%20tekoaly.pdf?sequence=1&isAllowed=y>. Luettu 25.3.2021
- 5 Packt. Surface automation with region mode. Verkkoaineisto. <https://subscription.packtpub.com/book/business/9781789610444/6/ch06lvl1sec53/surface-automation-with-region-mode>. Luettu 25.3.2021.
- 6 Global Payroll Association. What the History of RPA Technology Says About its Future. Verkkoaineisto. <https://globalpayrollassociation.com/blogs/technology/what-the-history-of-rpa-technology-says-about-its-future>. Luettu 25.3.2021.
- 7 Sogeti. Katsaus RPA-markkinoihin ja Blue Prism- ja UiPath-työkaluihin. Verkkoaineisto. <https://www.sogeti.fi/media/blog-posts/katsaus-rpa-markkinoihin-ja-blue-prism--ja-uipath-tyokaluihin/>. Luettu 25.3.2021.
- 8 TechTarget. Blue Prism brings RPA technology to healthcare. Verkkoaineisto. <https://searchhealthit.techtarget.com/feature/Blue-Prism-brings-RPA-technology-to-healthcare>. Luettu 25.3.2021.
- 9 Built In. Blue Prism is raising \$130M to fuel new technologies. Verkkoaineisto. <https://www.builtinaustin.com/2019/01/24/blue-prism-130-million-funding>. Luettu 25.3.2021.
- 10 Blue Prism. Blue Prism Server. Verkkoaineisto. <https://bpdocs.blueprism.com/bp-6-10/en-us/helpBPServer.htm>. Luettu 25.3.2021.
- 11 Blue Prism. Blue Prism Documentation. Verkkoaineisto. <https://bpdocs.blueprism.com/bp-6-10/en-us/home.htm>. Luettu 25.3.2021.
- 12 W3. XPath. <https://www.w3.org/TR/xpath/>. Luettu 17.10.2021.
- 13 Stackoverflow. Blue Prism: Cannot Iterate through JSON items withing injected JS script. <https://stackoverflow.com/questions/52762998/blue-prism-cannot-iterate-through-json-items-withing-injected-js-script>. Luettu 17.10.2021.

- 14 Python. General Python FAQ. Verkkoaineisto. <https://docs.python.org/3/faq/general.html>. Luettu 25.9.2021.
- 15 Robocorp. A new option for web automation: Using the Robot Framework Browser library, based on Playwright. Verkkoaineisto. <https://robocorp.com/docs/development-guide/browser/playwright>. Luettu 25.9.2021.
- 16 Playwright. Getting started. Verkkoaineisto. <https://playwright.dev/python/docs/intro/>. Luettu 25.9.2021.
- 17 Eficode. Robot Framework: Past, Present and Future. Verkkoaineisto. <https://www.eficode.com/blog/en/blog/robot-framework>. Luettu 25.9.2021.
- 18 Robot Framework. Introduction. Verkkoaineisto. <https://robotframework.org/?tab=1#getting-started>. Luettu 25.9.2021.
- 19 Github. Robot Framework 4.0. Verkkoaineisto. <https://github.com/robotframework/robotframework/blob/master/doc/releasenotes/rf-4.0.rst>. Luettu 30.9.2021.
- 20 Robot Framework. Why choose Robot Framework? Verkkoaineisto. <https://robotframework.org/rpa/>. Luettu 30.9.2021.
- 21 Grand view research. Robotic Process Automation Market Size, Share & Trends Analysis Report By Type, By Service, By Application, By Deployment, By Organization, By Region, And Segment Forecasts, 2021 – 2028. Verkkoaineisto. <https://www.grandviewresearch.com/industry-analysis/robotic-process-automation-rpa-market>. Luettu 30.9.2021.
- 22 Leadership Finland. Ohjelmistorobotiikka mullistaa tietotyön. Verkkoaineisto. <https://www.leadershipfinland.fi/artikkelit/ohjelmistorobotiikka+mullistaa+tietotyon/>. Luettu 30.9.2021.
- 23 HAMK. Ohjelmistorobotiikka työtä tehostamassa. Verkkoaineisto. <https://unlimited.hamk.fi/yrittajyys-ja-liiketoiminta/ohjelmistorobotiikka-tyota-tehostamassa/#.YWGzg9pBz-g>. Luettu 30.9.2021.
- 24 AI Multiple. 12-Step Guide to Best Practices of RPA Implementation. Verkkoaineisto. <https://research.aimultiple.com/rpa-implementation/>. Luettu 30.9.2021.
- 25 MorganFranklin. RPA Success Lies in Selecting the Right Processes. Verkkoaineisto. <https://www.morganfranklin.com/insights/company-insight/rpa-success-lies-in-selecting-the-right-processes/>. Luettu 30.9.2021.
- 26 YTJ. Yrityshaku. Verkkoaineisto. <https://www.ytj.fi/index/yrityshaku.html>. Luettu 30.9.2021.
- 27 Loihde advisory. Datan laadun hallinta. Verkkoaineisto. <https://www.loihdeadvisory.fi/palvelut/datanlaatu/>. Luettu 30.9.2021.

- 28 Helpsystems. Achieving Event-Driven Automation with Automate Triggers. Verkkoaineisto. <https://www.helpsystems.com/resources/articles/achieving-event-driven-automation-automate-triggers>. Luettu 17.10.2021.
- 29 CAI. RPA Exception Handling: Be in Control or Be Controlled. Verkkoaineisto. <https://www.cai.io/articles/rpa-exception-handling/>. Luettu 17.10.2021.
- 30 Tutorialspoint. Blue Prism - Exceptions Handling. https://www.tutorialspoint.com/blue_prism/blue_prism_exceptions_handling.html. Luettu 17.10.2021.
- 31 CGI. Mitä on tekoäly? Verkkoaineisto. <https://www.cgi.com/fi/fi/mita-on-tekoaly>. Luettu 17.10.2021.
- 32 Sogeti. Teknologiat hyperautomaation takana ja käyttötapausesimerkkejä. Verkkoaineisto. <https://www.sogeti.fi/media/blog-posts/teknologiat-hyperautomaation-takana-ja-kayttotapausesimerkkeja/>. Luettu 17.10.2021.
- 33 Blue Prism. Blue Prism 7.0 Release Notes. Verkkoaineisto. <https://bpdocs.blueprism.com/bp-7-0/en-us/release-notes/rn-7.0.htm>. Luettu 17.10.2021.