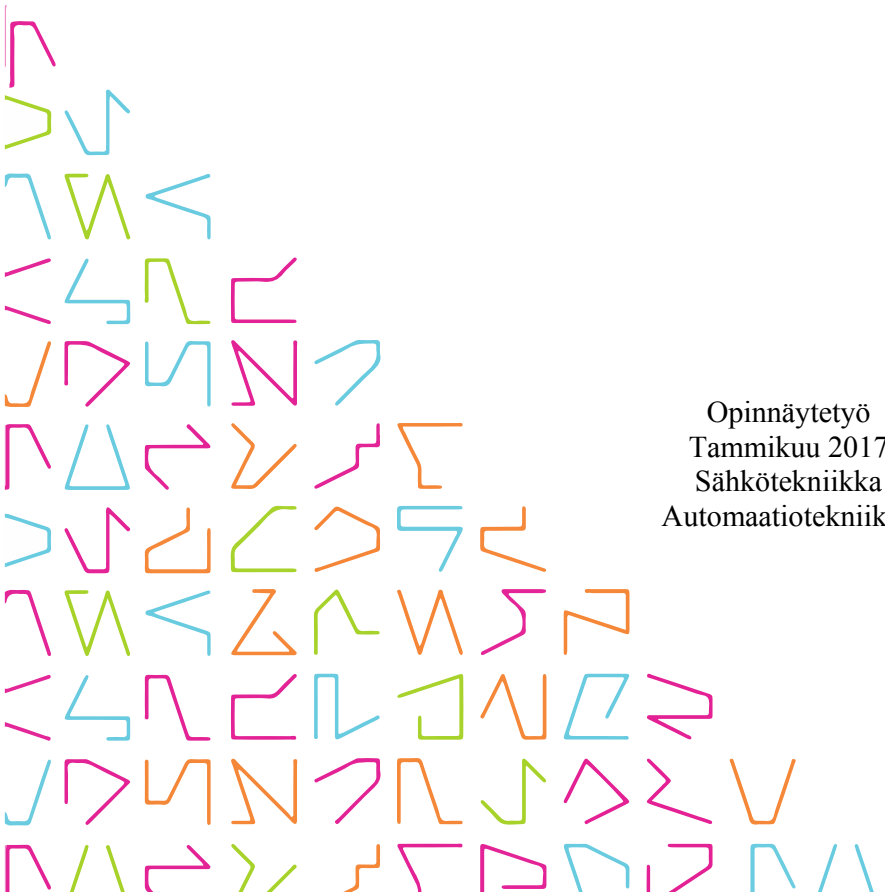
TAMPEREEN
AMMATTIKORKEAKOULU

KÄYTTÖOIKEUSHALLINNAN AUTOMATISOINTI OHJELMISTOROBOTIIKALLA

Juho Lindblad



Opinnäytetyö
Tammikuu 2017
Sähkötekniikka
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkötekniikan koulutusohjelma
Automaatiotekniikka

LINDBLAD, JUHO:
Käyttöoikeushallinnan automatisointi ohjelmistorobotiikalla

Opinnäytetyö 40 sivua, joista liitteitä 2 sivua
Tammikuu 2017

Tässä opinnäytetyössä kehitettiin automaatioratkaisu osaksi finanssialan käyttöoikeushallintajärjestelmän ylläpitoa. Käyttöoikeushallintajärjestelmän avulla hallitaan yrityksen työntekijöiden käyttöoikeuksia eri kohdejärjestelmissä. Automatisointiratkaisu toteutettiin ohjelmistorobotiikalla, ja sillä korvattiin aiemmin ulkopuoliselta kumppanilta ostettu palvelu. Automatisoinnilla tavoiteltiin nopeutunutta käsittelyaikaa sekä kustannussäästöjä. Työn tilaajana oli suomalainen finanssialan yritys.

Työ piti sisällään ohjelmistorobotiikkaan tutustumisen sekä varsinaisen toteutuksen suunnittelun, tarvittavan dokumentaation teon sekä käytännön toteutuksen. Prosessin automatisointi toteutettiin käyttäen Blue Prism -ohjelmistorobotiikan työkalua. Työn tavoitteena oli varsinaisen toteutuksen lisäksi perehtyä Blue Prismiin työkaluna sekä pohtia mallinnuksessa huomioitavia asioita.

Käyttöoikeusprosessin automatisointi onnistui odotusten mukaisesti, ja työssä kehitetty automaatioratkaisu otettiin tuotantokäyttöön joulukuussa 2016. Työn tilaaja oli tyytyväinen valmiiseen toteutukseen ja sen toteutusaikatauluun. Työllä saavutettu automaatioaste tehtävien käsittelyyn on 80–90 %. Automatisointi saatiin toteutettua ilman ulkopuolista apua, joten automatisoinnin takaisinmaksuajaksi saatiin noin 1–2 kuukautta.

Asiasanat: ohjelmistorobotiikka, Blue Prism, ohjelmointi, digitalisaatio

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical Engineering
Option of Automation Engineering

LINDBLAD, JUHO:
Automatizing Access Rights Management using Robotic Process Automation

Bachelor's thesis 40 pages, appendices 2 pages
January 2017

The purpose of this Bachelor's thesis was to create automation solution as part of access rights managing system for company in financial sector. Access rights managing system is used to control company's employee's access rights in different systems. Automatizing was done using Robotic Process Automation to replace the previous service bought from outsourcing partner. The goal was to reduce lead time and achieve cost savings. The work was ordered be a Finnish company from financial sector.

The thesis includes becoming acquainted with Robotic Process Automation and designing, documenting and creating the robotic solution. The business process was automatized using the Blue Prism Robotic Process Automation tool. The goals were to create the robotic solution, learn using Blue Prism software and detect the best practices that need to be taken into consideration during the modelling work.

The automatizing of the access right managing process was successful and the robotic solution was taken into production in December 2016. The client was satisfied with the final implementation and implementation schedule. 80–90 % of the tasks are now done using the Robotic Process Automation solution. Automatizing was successfully done without any external support, which resulted as short payback time of 1–2 months.

Key words: robotic process automation, Blue Prism, programming, digitalization

SISÄLLYS

LYHENTEET JA TERMIT

AI	Artificial Intelligence, tekoäly
Application modeller	Blue Prismin visuaalisen käyttöliittymän mallinnustyökalu
Blue Prism	Englantilainen RPA-yritys ja -ohjelmisto
BPO	Business Process Offshoring, Tietotyön prosessien ulkoistaminen
Business object	Blue Prismin uudelleenkäytettävä kokonaisuus, joka sisältää johonkin ohjelmaan liittyviä toiminnallisuuksia
CRM	Customer Relatioship Management, Asiakkuudenhallinta
C#	C Sharp – Ohjelmointikieli
J#	J Sharp – Ohjelmointikieli
Mainframe	Keskuskoneella toteutetut sovellukset, jotka ovat usein merkkipohjaisia
MAPIEx	Blue Prism ohjelmiston sähköpostiliitännäinen
Object studio	Blue Prism ohjelmiston objektitason ohjelmointiosa
PDD	Process Definition Document
Process studio	Blue Prism ohjelmiston prosessitason ohjelmointiosa
ROI	Return of Investment, Pääoman tuottoaste investoinnissa
RPA	Robotic Process Automation, Ohjelmistorobotiikka
SDD	Solution Definition Document
TAMK	Tampereen ammattikorkeakoulu
UAT	User Acceptance Test, Hyväksymistestaus
VB	Visual Basic – Ohjelmointikieli

1 JOHDANTO

Opinnäytetyön tarkoituksena on tutustua ohjelmistorobotiikkaan ja sen mahdollisuuksiin tietotyön prosessien tehostamisessa. Lisäksi tarkoituksena on suunnitella ja toteuttaa automaatioratkaisu suomalaisen finanssialan yrityksen käyttöoikeushallinnan tehostamiseksi. Käyttöoikeusprosessilla hallinnoidaan yrityksen työntekijöiden käyttöoikeuksia eri kohdejärjestelmiin.

Tietotyön prosessien tehostaminen on osa yrityksen digitalisaatiohanketta, jolla pyritään parantamaan asiakastyytyvyyttä sekä saavuttamaan kustannussäästöjä ja pienentämään käsittelyn läpimenoaikaa. Kehitystyössä päätettiin käyttää ohjelmistorobotiikkaa, koska se mahdollistaa tehtävien automatisoinnin ilman sovelluskehitystä. Prosessin automatisointi toteutettiin käyttäen Blue Prism -ohjelmistorobotiikan sovellusta.

Työ sisältää yleisen perehtymisen tietotyön tehostamiseen ja ohjelmistorobotiikkaan. Lisäksi työssä tutustutaan ohjelmistorobotiikan mallinnuksessa huomioitaviin asioihin sekä Blue Prismiin työkaluna. Työ sisältää myös varsinaisen toteutuksen suunnittelun, tarvittavan dokumentaation teon sekä käytännön toteutuksen ja testauksen. Tavoitteena on valmis ohjelmistorobotiikan sovellutus, jolla tilaajayritys saa kasvatettua käyttöoikeushallinnan automaatioastetta sekä nopeutettua valtuuskäsittelyn läpimenoaikaa.

Ohjelmistorobotiikka on suhteellisen uusi prosessinkehityksen muoto, jolla saadaan toteutettua automaatioratkaisuja tietotyön rutiiniprosesseihin. Ohjelmistorobotiikassa robotille mallinnetaan ihmisen käyttämät työpöytäsovellukset ja niiden käyttöliittymät sekä muodostetaan liiketoiminnalliset säännöt, joiden perusteella robotti toimii. Tämä mahdollistaa tehtävien automatisoinnin myös tilanteissa, joissa käytetään useampaa sovellusta, joiden keskinäinen integraatio ei ole toteutettavissa perinteisellä ohjelmistokehityksellä.

2 OHJELMISTOROBOTIIKKA OSANA TYÖN TEHOSTAMISTA

Ohjelmistorobotiikka on uusi ja hyvässä nosteessa oleva teknologia, jonka potentiaalia toiminnan tehostamisessa suuryritykset ovat testanneet hyvällä menestyksellä. Ohjelmistorobotti on sovellus, joka opetetaan jäljittelemään ihmisen toimintaa, mallintamalla sille käytettävien työpöytäsovellusten visuaaliset käyttöliittyvät sekä muodostamalla säännöt standardoidun prosessin työnkululle. Koska ohjelmistorobotti käyttää samaa käyttöliittymää kuin ihminen, sovellusten rajapintoihin tai lähdekoodiin ei tarvitse koskea, jolloin IT-hankkeiden luonne säilyy kevyenä. Roboteille hyvin soveltuvia tehtäviä ovat toistoa vaativat yksinkertaiset rutiinityöt. (Tamminen 2016)

2.1 Robotiikka osana prosessikehitystä

Kiristynyt kilpailutilanne ja asiakkaiden kasvaneet odotukset ovat ajaneet suuryritykset tehostamaan tietotyön prosessejaan. Prosessien tehostuksella pyritään saavuttamaan kustannussäästöjä sekä parempaa asiakastyytyväisyyttä. Tehostusprosessi on aiemmin jaettu viiteen eri tasoon:

1. Keskitä (centralize)
    - Palvelutuotannon keskittäminen fyysisesti samaan yksikköön
2. Standardoi (standardize)
    - Prosessien standardointi yksiköiden välillä
3. Optimoi (optimize)
    - Prosessien optimointi, virheiden ja turhan tekemisen minimointi
4. Uudelleen sijoita (relocate)
    - Keskitetyn palvelutuotannon siirtäminen halvemman työkustannusten maihin
5. Teknologistuminen (technology enable)
    - Palveluiden siirto verkkoon esimerkiksi asiakkaiden itsepalvelu-sovelluksiin

Näiden jatkoksi on viimeisen parin vuoden aikana muodostunut kuudes taso, automatisointi. Automatisoinnilla tarkoitetaan tässä yhteydessä laajemmalti eri

tekniikoita, joita ovat esimerkiksi RPA ja AI. RPA tulee sanoista Robotic Process Automation, joka tarkoittaa suomennettuna ohjelmistorobotiikka. AI on lyhenne sanoista Artificial Intelligence, eli suomennettuna tekoäly. Tekoälyllä tarkoitetaan tietokoneohjelmaa, joka kykenee älylliseen päättelyyn. Tekoälyn osana voidaan pitää myös kognitiivista eli oppivaa robotiikkaa, jonka tila on vasta kehitysvaiheessa. Ohjelmistorobotiikkakin on alkanut tekemään laajemmalti tuloaan vasta vuoden 2016 aikana. (Willcocks & Lacity 2016, 45–50)

Automatisoinnin avulla osa yritysten tietotyöstä siirtyy nyt keskitetyistä palveluntuotannoista konesaleihin. Outsourcing World Summit 2015 tapahtumassa järjestetyssä kyselyssä kolmannes vastaajayrityksistä arvioi, että yli puolet heidän palveluistaan soveltuu automatisoinnille. Yli neljänneksen automatisointiin uskoi 81 % prosenttia yrityksistä (Willcocks & Lacity 2016, 50–55). Automaatioasteen kasvaessa ei työvoimakustannuksilla ole enää niin suurta eroa kotimaan ja halvemman työvoiman maiden välillä. Tällöin asiantuntevuutta vaativat tehtävät säilyvät kotimaassa ja suurimman muutoksen kokevat halvemman työvoiman maat, joihin nykyisellään suuryritykset ovat ulkoistaneet taustapalveluitaan. RPA & Artifical Intelligence Summitin 2016 tekemässä kyselyssä 82 % vastaajista uskoi voivansa kehittää omia ratkaisuja RPA:n avulla, joilla olisi vaikutusta nykyiseen ulkoistamiseen (Barton 2016).

OpusCapitan Ventures-yksikön johtaja Petri Karjalainen pitääkin ohjelmistorobotiikkaa suurena mahdollisuutena Suomelle (Kolehmainen 2015). Hänen mielestään sen ansiosta Suomeen voi palata työpaikkoja, jotka ovat jo siirtyneet edullisemman työkustannuksen maihin, kuten Intiaan ja Puolaan. On kuitenkin selvää, että kehittyvän automaation ja digitalisaation myötä ihmisten työtehtävät tulevat muotoutumaan uudelleen myös kotimaassa. Elinkeinoelämän tutkimuslaitoksen julkaisemassa raportissa arvioidaan kolmasosan Suomen työpaikoista olevan uhattuna seuraavan kahdenkymmenen vuoden aikana (Pajarinen & Rouvinen 2014). Raportissa mainitaan muutoksen kohdistuvan esimerkiksi taloushallinnon työtehtäviin, kuten kirjapitoon ja palkanlaskuun. Nobel voittaja, professori Bengt Holmström on kuvannut digitalisaation murrosta seuraavasti: "Internet, robotit ja liiketoiminnan digitalisoituminen ovat kuin luonnonlakeja. Ei niiden kehittymistä voi estää. Mitä enemmän kiellämme uusia liiketoimintamalleja, sitä enemmän jäämme jälkeen" (Sajari 2015). Suomen täytyykin olla valppaana, jotta pystymme hyödyntämään maamme teknisten osaamisen ja pysymme kehityksen aallonharjalla.

2.2 Ohjelmistorobotiikan hyödyt

Ohjelmistorobotiikka on työkalu, jolla pyritään tuomaan prosessinkehitys tavallisista IT-hankkeista lähemmäksi loppukäyttäjiä. Tämä mahdollistaa myös pienempien kokonaisuuksien toteutuksen. Kehitystyökalut ovat tehty yksikertaisiksi, jotta niiden peruskäytön opettelu olisi mahdollista myös henkilöiltä, joilla ei ole pitkää ohjelmointitaustaa. Tähän on pyritty tekemällä ohjelmoinnista "mallinnusta", korvaamalla koodi valmiilla lohkoilla.

Ohjelmistorobotiikan etuna on kehityksen ketteryys, jolla saavutetaan kustannustehokkaita ja nopeasti toteutettavissa olevia automatisointeja. Tavoitteena ovat kustannussäästöt sekä parantunut tuottavuus ja laatu. Työntekijöiden tuottavuutta saadaan parannettua, kun heidän aikansa saadaan kohdennettua asiakkaille enemmän arvoa tuottaviin tehtäviin, kuten asiakaspalveluun ja asiantuntijatehtäviin. Robotti-ratkaisua käytettäessä, sen toimintalogiikka pysyy aina samana, jolloin kaikki tehtävät tehdään samalla tavalla ja inhimillisten virheiden riski poistuu. Virheettömyys ja nopeutunut käsittelyaika näkyvät parantuneena asiakastyytyväisyytenä sekä turhien yhteydenottojen ja reklamaatioiden vähenemisenä. (Tamminen 2016.)

Lisäksi yrityksen käyttämien järjestelmien elinkaarta saadaan pidennettyä, kun sovellusten keskinäisten integraatioiden puutetta saadaan paikattua ohjelmisto-robotiikalla. Kuitenkaan ohjelmistorobotiikka ei tulisi kuitenkaan koskaan nähdä pysyvänä ratkaisuna, vaan pidemmällä aikavälillä kehityksessä tulisi pyrkiä kohti suoraa automaatiota järjestelmäkehityksen avulla.

Suuryritykset ovat saaneet hyviä tuloksia toteutusten implementointikustannusten takaisinmaksuajoista. Esimerkiksi Britanniassa Telefónica O2 teleoperaattorin kerrotaan automatisoineen 15 ydinprosessiaan, jotka sisältävät noin puolimiljoonaa tehtävää kuukaudessa. Esimerkkejä heidän robotisoimistaan prosesseista ovat asiakkaiden SIM-korttien vaihdot sekä asiakkaiden prepaid-tilien saldolataukset. Aiemmin nämäkin toimenpiteet tehtiin manuaalisesti ja tehtävien tekeminen oli ulkoistettu intialaiselle kumppanille. Yhteensä näiden 15 prosessin kerrotaan säästävän satoja henkilötyövuosia ja niiden takaisinmaksuajaksi on laskettu 12 kuukautta. Kolmen vuoden mittarilla implementoinnin ROI eli investoinnin tuottoaste on 650–800 %. (Willcocks & Lacity 2016, 84–86.)

2.3 Prosessien edellytykset

RPA soveltuu yksinkertaisten ja määrämuotoisten prosessien automatisointiin. Nykyisellään RPA-robottien toiminta ei sisällä päättelyä tai harkinnanvaraisuutta, vaan sen toiminta on täysin sääntöpohjaista. Optimitilanteita ovat prosessit, joissa tietoa siirretään manuaalisesti usean eri järjestelmän välillä. Ohjelmistorobotiikan avulla tietoa saadaan siirrettyä kohdejärjestelmien välillä, vaikka niiden keskinäinen integraatio ei olisikaan toteutettavissa.

Toteutuksen mahdollisuuden kannalta on kriittistä, että käytettävä data on digitaalisessa ja strukturoidussa muodossa. Esimerkiksi robotin lukiessa tietoa pdf-tiedostosta, hyvässä tapauksessa pdf-tiedostoissa on ainoastaan määrämuotoista tietoa, joka on helposti eriteltävissä, kun taas huonossa tapauksessa tiedosto sisältää vapaamuotoista tai käsin kirjoitettua tekstiä.

Toteutuksen kannattavuuden kannalta on tärkeää, että prosessin tapahtumien volyymi sekä niiden tekemiseen käytettävä aika ovat riittävän suuret. Prosessin tulisi myös sisältää mahdollisimman vähän poikkeustapauksia, jolloin automaatioaste saadaan mahdollisimman korkeaksi.

Prosessin implementoinnin kestoon vaikuttaa edellä mainittujen asioiden lisäksi prosessin monimutkaisuus sekä valmiina olevan objektikirjaston laajuus mallinnettavien ohjelmistojen osalta. Ketterän ja nopean mallinnuksen kannalta onkin erittäin tärkeää, että kaikki objektit luodaan siten, että ne ovat helposti uudelleen käytettävissä.

3 BLUE PRISM-YHTIÖ JA -OHJELMISTO

Blue Prism on englantilainen ohjelmistorobotiikkaan erikoistunut yhtiö, joka kehittää suuryritysten käyttöön tarkoitettua RPA-ohjelmistoa. Blue Prism on tällä hetkellä markkinajohtajan asemassa, ja se onkin ainoa ohjelmistorobotiikan yritys, joka on listautunut pörssiin (Business Wire 2016). Sen asiakaskunta koostuu pääasiassa suurkokoisista pankeista, vakuutusyhtiöistä, teleoperaattoreista sekä julkishallinnon ja terveydenhoitoalan yrityksistä.

Blue Prsim -ohjelmisto tarjoaa suuryritysten käyttöön soveltuvan robotiikka-alustan. Se pitää sisällään työkalut ohjelmistorobottien kehittämiseen sekä robottityövoiman hallintaan ja ylläpitoon. Tässä luvussa keskitytään pääasiassa kehitystyökalujen esittelyyn. Kehitystyökalut koostuvat prosessitasolla Process Studiosta ja objektitasolla Object Studiosta.

Blue Prismin avulla voidaan kehittää ohjelmistorobotteja, jotka käyttävä Windowsin työpöytäsovelluksia, Java-pohjaisia sovelluksia, selainpohjaisia sovelluksia (html) sekä merkkipohjaisia Mainframe-sovelluksia. Lähtökohtaisesti sovellus käyttää eri kenttiä ja painikkeiden sovelluksessa olevia parametreja sekä taulukkorakenteita. Visuaalisen mallinnuksen tukena on mahdollista käyttää Microsoftin kehittämää .NET Framework ohjelmistokomponenttikirjastoa C#-, J#- ja VB-ohjelmointikielillä. Lähtökohtaisesti mallinnus on kuitenkin toteutettavissa pelkällä olio-ohjelmoinnilla sekä Blue Prismin tarjoamilla valmiilla koodikokonaisuuksilla. Liiketoimintatyötä hoitavista mallinnus-kokonaisuuksista käytetään nimityksiä prosessi ja ohjelmistorobotti.

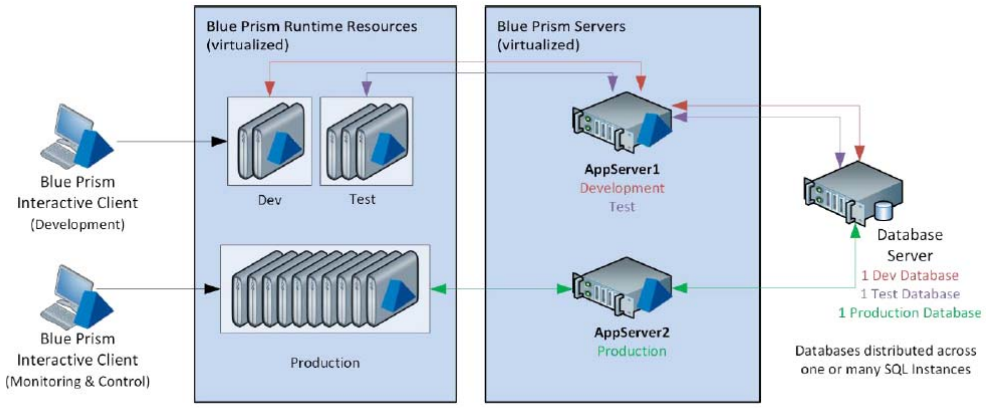
3.1 Blue Prism -ympäristö

Blue Prism -ympäristö on muodostettu vastaamaan suuryritysten tarpeita. Sen laajuus on helposti skaalattavissa yrityksen tarpeen mukaan. Blue Prism -ympäristö koostuu seuraavista osista (Blue Prism Infrastructure Overview Data Sheet 2015):

- Blue Prism Interactive Client
    - Fyysinen tai virtuaalinen kone, jota käytetään Blue Prism -prosessien kehittämiseen, määritykseen ja ylläpitoon

- Kopio tavallisen käyttäjän työpöydästä, johon on asennettu tarvittavat työpöytäsovellukset sekä Blue Prism -sovellus

- Blue Prism Runtime Resource PC
  - Fyysinen tai virtuaalinen kone, jolla suoritetaan automatisoituja Blue Prism -prosesseja automaattisella käynnistyksellä
  - Kopio tavallisen käyttäjän työpöydästä, johon on asennettu tarvittavat työpöytäsovellukset sekä Blue Prism sovellus
- Blue Prism Application Server
  - Fyysinen tai virtuaalinen kone, jolle on asennettuna Blue Prism Server
  - Muodostaa suojatusti yhteyden tietokannan ja clienttien välillä sekä ylläpitää aikataulua, jolla ohjataan Runtime Resourceja
- Blue Prism Database
  - SQL Server -tietokanta, jossa säilytetään keskitetysti prosessit, käyttäjätiedot ja lokit
  - Yleensä kehitys- ja tuotantoympäristöille on omat tietokantansa



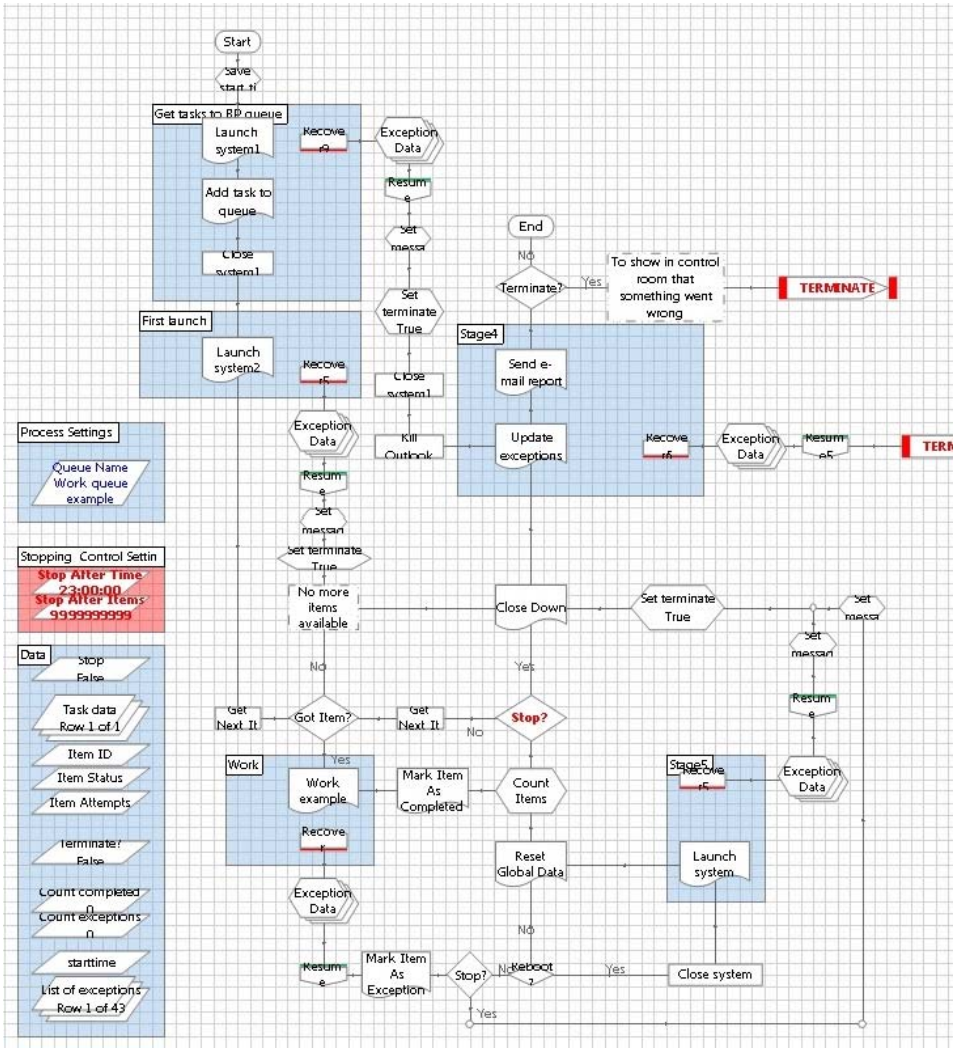
KUVA 1. Esimerkki Blue Prism –ympäristöstä (Blue Prism Infrastructure Overview Data Sheet 2015)

Kuvassa 1 on kuvattu esimerkki mahdollisesta ympäristöjaosta. Normaalisti kehitys-, testaus- ja tuotantotietokannat ovat erotettuna toisistaan. Kehitysympäristö (Dev) on ympäristö, jossa tehdään uusien robotiikkaprosessien kehitystä. Testausympäristö (Test) on ympäristö, jota voidaan käyttää testaukseen kehitystyön valmistuttua ennen tuotantoon siirtoa. Kehittäjillä on pääsy käyttämänsä koneen Interactive Clientin kautta kehitys- ja testausympäristöihin, jotka ovat yhteydessä Application Serverin kautta ympäristöille määritettyihin tietokantoihin.

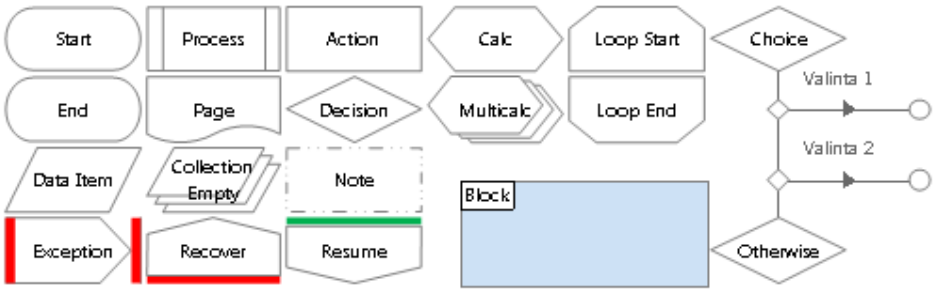
Tuotannon tietokanta ja Application Server, ovat erotettuna kehityspuolesta. Niihin on pääsy ainoastaan ylläpitoon tarkoitettujen Interactive Clientien välityksellä. Ympäristöjen ja tietokantojen erottamisella taataan tuotannon toimivuus. Tällöin tuotantoympäristössä on ainoastaan testattuja ja hyväksyttyjä prosesseja, eikä kehityspuolella tehdyt muutokset vaaranna tuotannossa olevia prosesseja. Ennen uusien prosessien siirtoa tuotantotietokantaan tulee varmistaa, ettei niistä aiheudu haittaa jo tuotantoajossa oleville robotiikkaprosesseille ja että prosessit täyttävät yrityksen asettamat laatu- ja tietoturvavaatimukset.

3.2 Process Studio -työkalu

Process Studio -työkalulla mallinnetaan liiketoimintaprosessin työnkulku. Tämän visuaalinen näkymä muistuttaa tavallisten työnkulun mallinnustyökalujen, kuten MS Vision, näkymää. Process Studiossa kuvataan vastaavalla tavalla eri toimintoja symbolien avulla. Tässä näkymässä muodostetaan visuaalista käyttöliittymää hyödyntäen robottiohjelman koodi kytkemällä peräkkäin ohjelmakokonaisuuksia. Kuvassa 2 on esitelty esimerkkinäkymä prosessin etusivusta.



KUVA 2. Esimerkki prosessin etusivusta



KUVA 3. Process Studio -työkalun lohkot

Kuvassa 3 on esiteltynä Process Studio -työkalun lohkot, joiden avulla prosessin kulkua mallinnetaan. Seuraavassa luettelossa on kerrottu kunkin lohkon toiminta.

- START = Sivun käynnistyslohko, josta suoritus alkaa. Voidaan vastaanottaa tietoa Input-parametrien avulla.
- END = Sivun lopetuslohko. Suoritus palaa kohtaan josta kyseistä sivua tai toimintoa on kutsuttu. Voidaan palauttaa tietoa Output-parametrien avulla.
- PROCESS = Prosessilohko, jonka avulla voidaan kutsua toista prosessia.
- PAGE = Sivulohko, jonka avulla voidaan kutsua saman prosessin toista sivua.
- ACTION = Toimintolohko, jonka avulla voidaan kutsua Object Studiossa määritettyä toiminnallisuutta.
- DECISION = Valintalohko, jonka avulla prosessin kulku voidaan jakaa kahteen vaihtoehtoiseen polkuun. Ehdon toteutuessa jatketaan "Yes"-polkuun ja ehdon ei toteutuessa "No"-polkuun. Vastaa ohjelmoinnin IF-lausetta.
- CHOICE = Valintalohko useammalla ulostulolla. Mikäli yksikään ehdoista ei täyty, jatketaan suoritusta "Ohterwise"-kohdasta. Vastaa ohjelmoinnin ELSE IF -lausetta.
- CALC = Laskulohko, jonka sisällä suoritetaan laskutoimituksia esimerkiksi numeroille, merkkijonoille tai päivämäärille.
- MULTICALC = Laskulohko usealla laskutoimituksella. Vastaa peräkkäin ketjutettuja CALC-lohkoja.
- LOOP = Silmukkalohko. Käytetään saman toiminnallisuuden toistamiseen. Käyttää hyödyksi taulukoiden (Collection) arvoja. Vastaa ohjelmoinnin FOR-silmukkaa.
- DATA ITEM = Muuttuja, johon voidaan tallentaa tietoa. Muuttujille voidaan määrittää oletusarvot ja niihin voidaan tallentaa arvoja prosessin aikana. Luettelo eri muuttujista:
    - Date = Päivämäärä
    - DateTime = Päivämäärä ja kellonaika
    - Flag = Bittimuuttuja, jonka arvo on True tai False
    - Number = Integer- eli numeromuuttuja
    - Text = String- eli merkkijonomuuttuja
    - Password = Merkkijonomuuttuja, jonka sisältö on salattu
    - Time = Aikamuuttuja, joka kuvastaa kellonaikaa
    - TimeSpan = Aikamuuttuja, joka kuvastaa kulunutta aikaa
    - Image = Kuva
    - Binary = Binäärimuuttuja
- COLLECTION = Taulukko, joka koostuu Data itemeiden arvoista

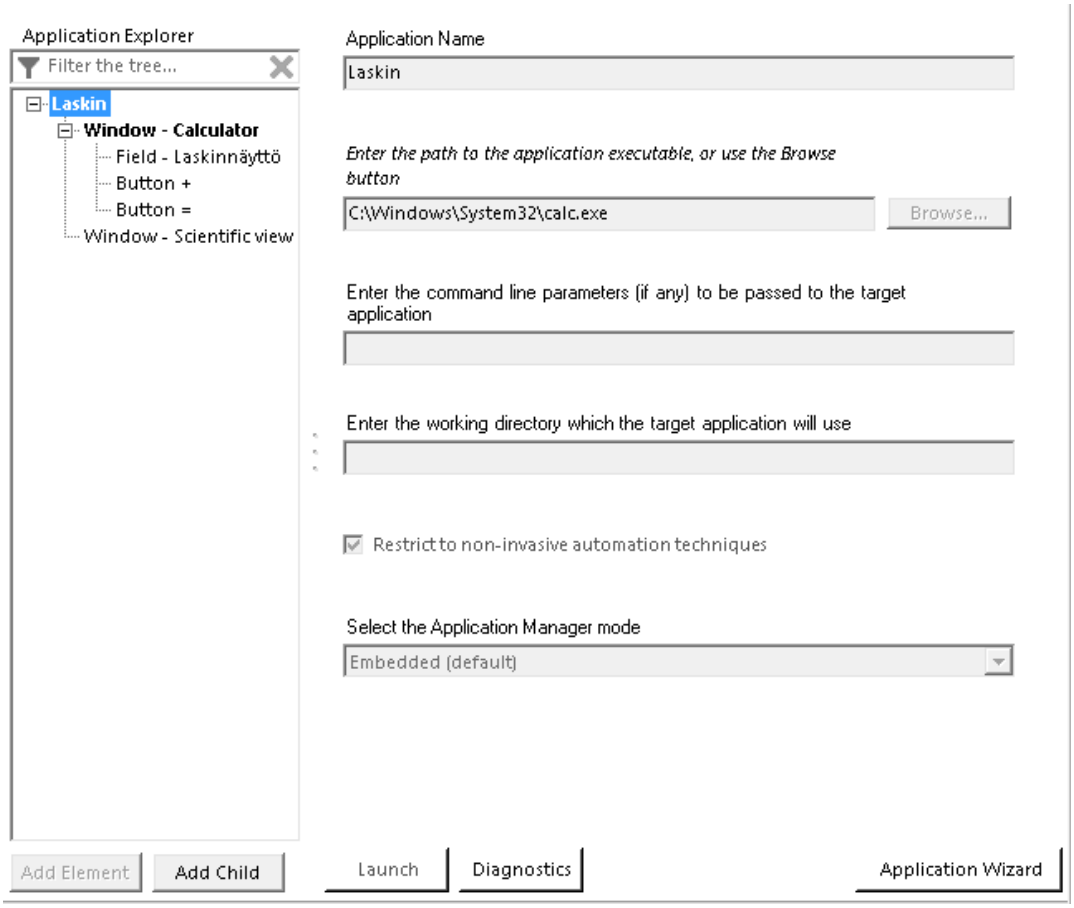
- NOTE = Kommenttilohko, joka ei vaikuta prosessin toiminnallisuuteen.
- EXCEPTION = Poikkeuslohko, jonka avulla prosessi voidaan ajaa poikkeustilaan.
- RESUME = Jatkumislohko, jonka avulla prosessi voidaan palauttaa poikkeustilasta normaalitilaan.
- RECOVER = Palautuslohko, josta prosessin suoritus jatkuu poikkeustilanteessa.
- BLOCK = Lohkotyökalu, jonka avulla prosessia voidaan ryhmitellä. Esimerkiksi poikkeustilan syntyessä lohkon sisällä, jatkuu käsittely ensisijaisesti lohkon sisällä olevasta Recover-lohkosta.

Prosessi muodostetaan käyttäen edellä mainittuja lohkoja. Tarkoituksena on että prosessitasolla tehdään kaikki liiketoimintaprosessiin liittyvä päättely ja Action-lohkoilla kutsuttavilla toiminnallisuuksilla, eli objekteilla, suoritetaan haluttuja toimintoja kohdejärjestelmissä.

3.3 Object Studio -työkalu

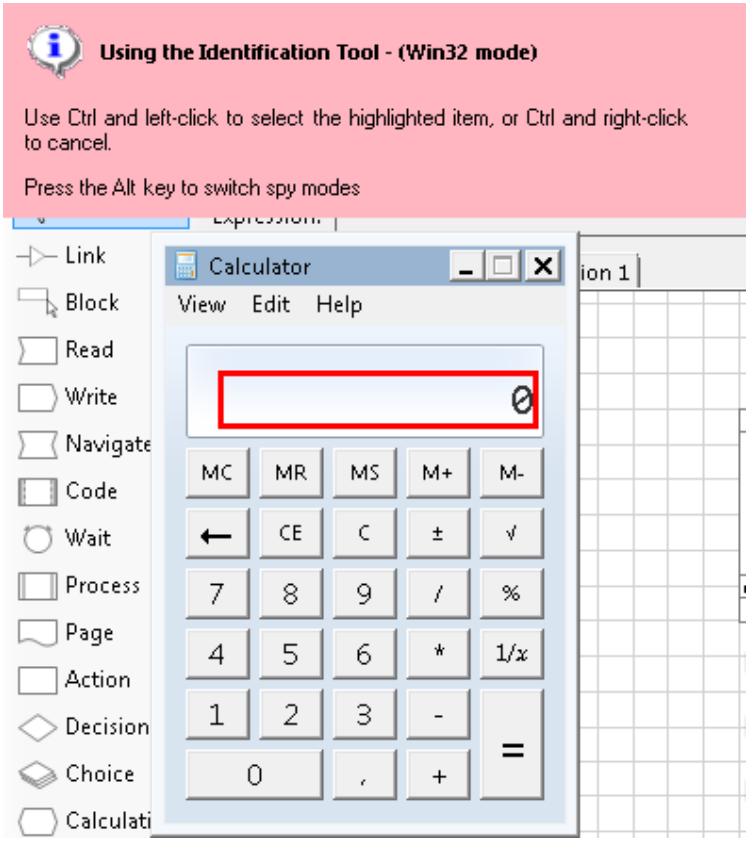
Object Studio on Blue Prismin työkalu, jolla voidaan luoda uudelleen käytettävissä olevia ohjelmakokonaisuuksia, Business Objecteja. Business Objectit ovat aina sovelluskohtaisia ja sisältävät ohjelman visuaalisen mallinnuksen sekä niiden pohjalta rakennettuja uudelleen käytettäviä toiminnallisuuksia, eli Actioneita. Business Objecteja ei koskaan käytetä sellaisenaan, vaan niitä kutsutaan aina prosessitasolta. Objektien ja prosessien välillä voidaan siirtää tietoa määrittämällä objektin Start- ja End-lohkoihin Input- ja Output-muuttujia. Objektien käyttö perustuu pitkälti sovellusten käyttöliittymien mallinnukseen, joka tehdään Application Modeller -työkalulla.

Application Modeller on Object Studion osa, jolla objekti liitetään haluttuun sovellukseen ja sen osiin. Siellä määritetään sovelluksen käynnistysasetukset sekä sovellusten eri osiot tunnistetaan tunnistustyökalun ja siihen liitettävien parametrien avulla. Application Wizard -toiminnon avulla voidaan määrittää sovelluksen tiedot, kuten sovellustyypin, käynnistystiedoston sijainnin tai verkko-osoitteen (Kuva 4). Application Modellerin vasemmassa reunassa näkyy puurakenne mallinnetuista ikkunoista ja toiminnoista.



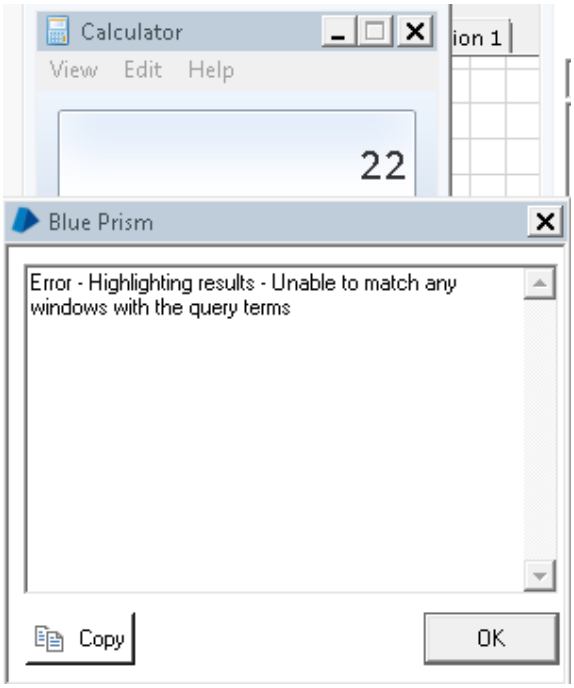
KUVA 4. Application Modeller

Uuden kohteen eli elementin tunnistus tehdään Identify-toiminnolla. Käytettävät tunnistusmenetelmät vaihtelevat kohdejärjestelmän sovellustyypin mukaan. Lähtökohtaisesti Blue Prism käyttää elementin tunnistuksessa sovelluksen lähdekoodista saatavia tietoja. Esimerkkinä kuvassa 5 käytetään Win32 mode -tunnistustyökalua, jolla tunnistetaan Windowsin laskimen numerokenttää. Käytettävä tunnistustyökalu vaihtelee mallinnettavan sovelluksen mukaan. Mikäli sovelluksen kentät eivät ole tunnistettavissa millään työkalulla, voidaan viimeisenä vaihtoehtona käyttää kuvatunnistusta eli Region mode -tunnistustyökalua. Kuvatunnistuksessa ohjelmiston käyttö voidaan mallintaa ottamalla kuvia esimerkiksi käytettävistä painikkeista. Tekstejä voidaan tunnistaa kuvista määrittämällä käytettävän fontin tiedot tai hyödyntämällä Googlen Tesseract OCR (Optical Character Recognition) -työkalua. Kuvatunnistuksen avulla mallintamista tulee käyttää vasta viimeisenä vaihtoehtona, sillä mallinnus on työläämpää ja vikaherkempää. Sovelluksen ulkoasu saattaa muuttua koneen asetuksia, kuten resoluutiota tai käytettävää teemaa, muutettaessa, mikä rikkoo objektien toimivuuden.



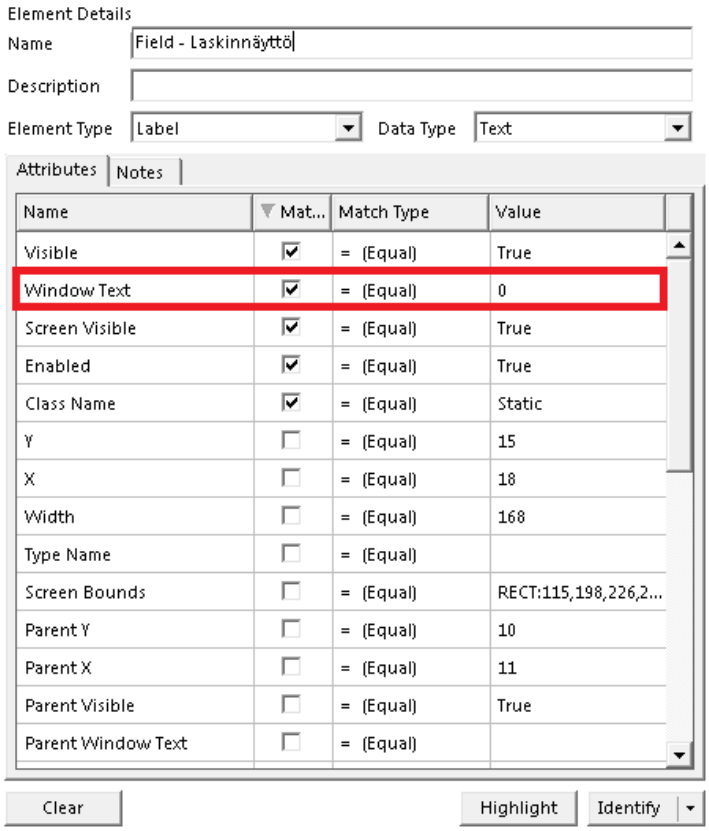
KUVA 5. Laskimen kentän tunnistaminen käyttäen Identification-toimintoa

Tunnistuksen jälkeen kohteen identifioinnin voi testata Highlight-toiminnolla. Tässä esimerkkitapauksessa Highlight-toimintoa käytettäessä ohjelma korostaa äsken identifioidun numerokentän. Kuitenkin mikäli laskimen kentän arvo "0" vaihdetaan toiseksi, Highlight-toiminto ei enää tunnista kohdetta. Kuvassa 6 kentän arvoksi on muutettu 22, joka estää kentän tunnistuksen.



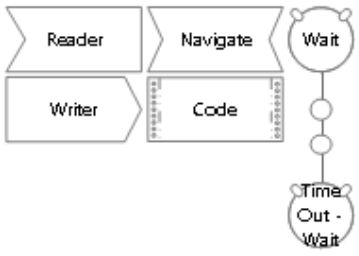
KUVA 6. Highlight-toiminnon virheilmoitus epäonnistuneesta korostuksesta

Ohjelma voidaan korjata tunnistamaan kohteen muuttamalla tunnistuksen parametreja. Tunnistustyökalu tuo usein oletuksena tietoja, jotka ovat ominaisia ainoastaan sen hetken tilanteelle. Tällaisia voivat olla esimerkiksi asiakkaaseen viittaavat yksilöidyt tiedot, kuten äskeisessä esimerkissä olevat muuttuvat kenttien arvot. Esimerkin tapauksessa kohteen parametreista tulee poistaa valinta "Window text", jolloin ohjelma jälleen tunnistaa kohteen (Kuva 7). Mikäli sovelluksessa olisi useampia samalla luokituksella (Class Name) olevia kenttiä, tulisi kohteen tunnistamiseksi ottaa käyttöön myös muita parametreja. Parametrien tarkistukset tulee huomioida esimerkiksi CRM-järjestelmissä, joissa esiintyy asiakkaaseen viittaavia tietoja, kuten asiakas- tai sopimustunnuksia. Tällöin voidaan käyttää Wildcard-toimintoa, jolla on mahdollista korvata osittain parametrin arvo. Esimerkiksi, jos sovelluksen ikkunan nimi on toisella sopimuksella ”Sopimus, 24125” ja toisella ”Sopimus, 53215”, tulee parametrin arvo korvata Wildcardilla ”Sopimus*”, jolloin nimen loppuosaa ei enää huomioida.



KUVA 7. Parametrinäkymä

Object Studio -työkalusta löytyvät samat toimilohkot kuin Process Studio -työkalustakin. Näiden lisäksi löytyvät kuvan 8 mukaiset lohkot. Lohkojen toiminta on kuvattu seuraavassa luettelossa.



Kuva 8. Object Studio -työkalun toimilohkot

- READER = Lukulohko, jolla voidaan lukea arvo Application Modeller -työkalulla mallinnetusta kentästä
- WRITER = Kirjoituslohko, jolla voidaan kirjoittaa arvo Application Modeller -työkalulla mallinnettuun kenttään

- NAVIGATE = Navigointilohko, jolla voidaan esimerkiksi aktivoida ikkunoita tai lähettää kohdejärjestelmään painikkeiden painalluksia tai näppäinkomentoja.
- CODE = Koodilohko, jolla voidaan ohjelmoida toiminnallisuuksia. Käytettävä ohjelmointikieli (C#, J# tai Visual Basic) valitaan objektin pääsivulla.
- WAIT = Odotuslohko, jota käytetään toimintojen välillä odottamaan toimintojen toteutumista, kuten esimerkiksi sovellusten ikkunoiden aukeamista. Lohkolle voidaan määrittää useampi mahdollinen odotettava elementti sekä Time Out -aika.

Objekteja luotaessa, tulee muodostaa mahdollisimman pieniä ja yleiskäyttöisiä kokonaisuuksia. Esimerkiksi yksi objekti voisi pitää sisällään toiminnallisuudet kohdejärjestelmän käynnistykseen ja sisäänkirjautumiseen. Tällöin oikein muodostettuna objektit toimivat suoraan kaikkien kyseistä järjestelmää käyttävien prosessien kanssa. Hyvin luotu objektikirjasto on tehokkaan työskentelyn edellytys, koska sillä saadaan ehkäistyä päällekkäisen työn tekemistä sekä parannettua robottiprosessien ylläpitoa. Kun kaikki prosessit käyttävät samaa objektia, voidaan samaan sovellusikkunaan kohdistuvat muutokset tehdä yhtä objektia muokkaamalla.

3.4 Blue Prismin työjonot

Blue Prism -ohjelmisto pitää sisällään oman työjonotoimintonsa, jota käytetään työtehtävien hallitsemiseen. Yksi prosessi voi käyttää useampaa työjonoa ja sama työjono voi olla useamman prosessin käytössä. Työjononäkymästä on nähtävissä kunkin työjonon tilannetiedot, kuten käsiteltyjen ja käsittelemättömien tehtävien kappalemäärät. Työjononäkymästä voidaan myös tarkistella kunkin yksittäisen tehtävän tietoja, kuten lisäysajankohtaa, työvaihetta tai poikkeustapauksen virhesyytä. Yksittäisten tehtävien tunnistamiseen käytetään työjonon asetuksissa määritettyä Key Name -valintaa. Sen avulla tehtävälle annetaan työjononäkymässä näkyvä yksilöivä nimitieto, kuten esimerkiksi asiakastunnus tai rekisterinumero. Työjonoon ladatut tiedot säilytetään salattuna Blue Prismin SQL-tietokannassa.

Työjonot mahdollistavat tehtävätietojen siirtämisen Blue Prismin sisään prosessin alkuvaiheessa. Usein lähtötietona on esimerkiksi Excel-tiedosto, jossa on useita satoja rivejä tehtävätietoja. Blue Prismin avulla tehtävätiedot erotellaan Excelistä

Collectioniksi, ja ladataan kerralla työjonoon. Latausvaiheessa voidaan tehtäville määrittää myös lisätietoja, kuten mahdollisia prioriteettitasoja tai aikalukituksia.

Prosessin suorittaessa työtehtäviä, poimitaan ne yksitellen työjonosta prioriteetti- ja aikatauluperusteisesti. Yhdellä prosessilla voi olla käsittelyssä ainoastaan yksi tehtävä samanaikaisesti. Tehtävän käsittelyn edetessä, voidaan sen Status- eli tilatietoa päivittää tehtävävaiheittain. Päivitetyn tilatiedon avulla voidaan tarvittaessa jatkaa keskeytyneen tehtävän käsittelyä oikeasta työvaiheesta. Kun tehtävä saadaan valmiiksi, kuitataan se valmiiksi työjonossa. Poikkeustapauksien kohdalla sama tehtävä voidaan myös lisätä työjonoon uudelleen yritettäväksi, mikäli työjonon asetukset ja poikkeustapauksen aiheuttanut virhesyy sen sallivat.

4 KÄYTTÖOIKEUSPROSESSI

Käyttöoikeusprosessilla hallitaan finanssialan yrityksen työtekijöiden käyttöoikeuksia eri kohdejärjestelmien osioihin. Joidenkin tilaajayrityksen sektoreiden käyttöoikeushallintaa on jo automatisoitu sovelluskehityksen ja sovellusten keskinäisten integraatioiden avulla. Täysi automaatio ei ole kuitenkaan ollut mahdollinen erään sektorin käyttöoikeushallinnassa, koska siinä käytettävään kohdejärjestelmään ei ole ollut mahdollista tai kustannustehokasta toteuttaa integraatiota. Prosessin alkuvaiheet oli jo pitkälti automatisoitu, mutta lopullinen käyttöoikeuksien lisäys ja poisto tehtiin edelleen manuaalisesti. Käyttöoikeudet on jaettu pieniin kokonaisuuksiin, minkä takia samalle henkilölle saatetaan hakea kerralla parikymmentäkin eri käyttöoikeutta. Prosessinkehitystä oli jo tehty tavoitteena kustannussäästöt sekä nopeampi läpimenoaika. Näihin tavoitteisiin oli pyritty ulkoistamalla käyttöoikeuksien lisäys ja poisto yhteistyökumppanille ja määrittämällä heille tiukahko palvelulupaus. Robotisoinnilla pyrittiin vielä nopeampaan hakemusten läpimenoaikaan, virhetilanteiden minimointiin sekä kustannussäästöihin.

Tässä luvussa on kuvattu prosessin eri vaiheet käyttöoikeuden tilaamisesta hakemuksen käsittelyyn. Työn tilaajan vaatimuksesta prosessia ei kuvata yksityiskohtaisesti, eikä järjestelmien nimiä julkaista.

4.1 Prosessi ennen ohjelmistorobottia

Käyttöoikeusprosessin pelkistetty prosessikaavio ennen ohjelmistorobottia löytyy liitteestä 1. Ensimmäisessä vaiheessa käyttäjä ja hänen esimiehensä hakee haluttua valtuutta tai valtuuden päättämistä selainpohjaisessa sovelluksessa. Siirtyäkseen käsiteltäväksi valtuushakemuksessa tulee olla esimiehen sekä sovelluskohtaisen valvovan tahon hyväksynnät. Tarvittavien hyväksyntöjen jälkeen hakemus siirtyy palvelupyyntönä palvelupyyntöjen kirjaus- eli tiketöintijärjestelmään odottamaan ulkoisen kumppanin käsittelyä. Tehtävän siirto on toteutettu sähköposti-integraation avulla. Siinä järjestelmä lähettää sähköpostiviestin, jonka sisältö ja otsikko ovat määrämuotoisia. Otsikossa olevien tietojen perusteella palvelupyyntö kirjautuu oikeaan paikkaan järjestelmässä. Hakemuksen mukana kulkee valtuuden, käyttäjän sekä tilaajan tiedot.

Ulkoisen kumppanin toimihenkilö vastaanottaa tehtävät tiketöintijärjestelmästä ja suorittaa valtuuden lisäyksen tai päättämisen hakemuksen mukaisesti kohde-järjestelmässä. Mikäli hakemukselta puuttuu tilauksen kannalta oleellisia tietoja, etsii toimihenkilö ne muista järjestelmistä. Kun valtuus on onnistuneesti käsitelty, kuitataan tiketti valmiiksi, jolloin lähetetään myös vahvistusviestit valtuuden tilaajalle ja saajalle.

Yhden tehtävän käsittelyyn menee arviolta noin 5–10 minuuttia. Tehtävien määrä viikossa on noin 300–400 kappaletta. Tehtävien määrä vaihtelee vuodenajoittain ja suurin piikki on alkukesästä.

4.2 Ohjelmistorobotin tuomat muutokset prosessin kulkuun

Käyttöoikeusprosessin pelkistetty prosessikaavio ohjelmistorobotin kanssa löytyy liitteestä 2. Valtuuden tilaukset ja hyväksynnät selainpohjaisessa tilausjärjestelmässä pysyvät ennallaan. Sovelluksen lähettämää viestiä ei enää ohjata suoraan tiketöintijärjestelmään, vaan se lähetetään sähköpostikansioon, johon ohjelmisto-robotille on lisätty pääsyoikeus. Käynnistyessään robotti käsittelee kansioon saapuneet sähköpostit. Viestistä robotti tarkistaa ensin lähettäjän sähköpostiosoitteen sekä viestin otsikon. Mikäli viesti on tullut väärästä osoitteesta tai mikäli sen otsikko ei täytä sille asetettuja kriteerejä, siirretään viesti muulle postille ositettuun roskapostikansioon. Mikäli edellä mainitut kriteerit täyttyvät ja tehtävä tunnistetaan validiksi, lukee robotti sähköpostin viestisisällön. Esikäsittelyyn on tehty kyseiset tarkistukset väärinkäytösten estämiseksi.

Robotti erottelee sähköpostiviestistä tarvittavat tiedot ja tunnistaa, onko kyseessä käyttäjätunnuksen luonti, valtuuden lisäys vai valtuuden poisto. Viestikentästä poimittavia tietoja ovat esimerkiksi käyttäjien nimet ja sähköpostiosoitteet sekä valtuuteen liittyvät tiedot, kuten sovelluksen nimi, valtuuden tunnus ja kuvaus. Osassa tehtävistä robotti joutuu etsimään puuttuvan käyttötunnuksen toisesta järjestelmästä. Kun kaikki tarvittavat tiedot on saatu poimittua, lisää robotti tiedot Blue Prismiin prosessille luotuun työjonoon ja siirtää sähköpostin keskeneräisten tehtävien kansioon. Lisätessä robotti myös priorisoi tehtävät niiden tehtävätyyppien mukaan. Mikäli hakemusviestistä ei löydy kaikkia tarvittavia tietoja, robotti ohjaa tehtävän

tiketöintijärjestelmään sähköposti-integraation avulla. Tiketöintijärjestelmästä tehtävä käsitellään manuaalisesti. Robotti käy läpi sähköpostikansiota, kunnes siellä ei enää ole viestejä.

Sähköpostikansion käsittelyn jälkeen robotti käynnistää kohdejärjestelmän ja kirjautuu sinne sisään omilla käyttötunnuksillaan. Seuraavaksi robotti navigoi sovelluksen sisällä käyttöoikeuksien hallintajärjestelmään. Robotti ottaa uuden tehtävän Blue Prismin työjonosta ja suorittaa tehtävätyypin edellyttämät toimenpiteet. Kun tehtävä on valmis, lähettää robotti sähköpostilla kuittauksen käyttäjälle ja valtuuden tilaajalle. Lisäksi robotti arkistoi alkuperäisen sähköpostin valmiiden tehtävien kansioon. Mikäli käsittelyssä tapahtuu jotain poikkeavaa, lähettää robotti tehtävän tiketöintijärjestelmään manuaalikäsittelyä varten. Kun työjono on saatu tyhjäksi, sulkee robotti kohde-järjestelmän ja lähettää loppuraportin valvovalle taholle.

5 TYÖVAIHEET

Tässä kappaleessa esitellään opinnäytetyön toteutuksen työvaiheet. Työvaiheita olivat esiarviointi, dokumentointi, mallinnus, testaus sekä tuotantoon siirto. Salassapidon vuoksi työvaiheita esitellään osittain esimerkkien kautta.

5.1 Esiarviointi

Ennen työn aloittamista liiketoimintaprosessille suoritettiin esiarviointi, jossa tutkittiin prosessin soveltumista ohjelmistorobotiikalle. Esiarvioinnissa selvitettiin muun muassa käytettävät kohdejärjestelmät, arvioidut kustannussäästöt, käytössä olevat prosessi-asiantuntijat sekä prosessiin mahdollisesti liittyvät riskit ja ongelmakohdat.

Kustannussäästöissä arvioidaan sekä välittömiä että välillisiä kustannussäästöjä. Välittömiä kustannuksia ovat esimerkiksi liiketoimintaprosessin sitomat henkilöstö-resurssit. Nämä määräytyvät tehtävien kappalemääristä ja käsittelyajoista. Tässä prosessissa suorat kustannussäästöt tulevat palvelupyyntöjen vähentymisenä. Välillisiä kustannussäästöjä saadaan käsittelyaikojen lyhentymisenä. Joidenkin prosessien kohdalla ohjelmistorobotin myötä nopeutunut käsittelyaika esimerkiksi vähentää yhteydenottoja asiakaspalveluun tai parantaa asiakastyytyvyyttä ja luo täten kilpailuetua. Käyttöoikeushallinnassa nopeutunut käsittelyaika mahdollistaa valtuuksien myöntämisen nopealla aikataululla myös ruuhkahuipuissa ja kiireellisissä tilanteissa.

Kohdejärjestelmien soveltuvuutta arvioitaessa tulee ottaa huomioon niiden soveltuvuus ohjelmistorobotiikan mallinnustyökaluille. Joidenkin sovellusten kohdalla voi olla robotin vaikea tunnistaa sovelluksen elementtejä, jolloin mallinnuksessa joudutaan käyttämään kuvatunnistusta. Tällöin työmäärä kasvaa merkittävästi, sekä lisäksi prosessista tulee vikaherkempi kohdejärjestelmän tai alustan muutoksille. Toinen arvioitava seikka on valmiina olemassa oleva objektikirjasto. Mikäli käytettäville sovelluksille ja niiden toiminnoille on jo olemassa valmiita objekteja, vähentää tämä mallinnuksen työmäärää. Tämän prosessin kohdalla käytettiin pääasiassa Mainframe-sovellusta, joka sopii ohjelmistorobotille hyvin.

Prosessiasiantuntijoiden käytettävyys on tärkeässä roolissa, koska ohjelmistorobotille tulee määrittää tarkasti tehtävien kulku ja liiketoiminnalliset säännöt, joiden pohjalta robotti toimii. Ilman prosessiasiantuntijan sitoutumista toimivan robottiohjelmiston tekeminen ei ole mahdollista. Tämän prosessin osalta prosessiasiantuntijoita oli käytettävissä.

Tämän prosessin osalta kaikki tarvittavat kriteerit täyttyivät, eikä mitään toteutusta estäviä seikkoja löytynyt, joten työ otettiin toteutettavaksi.

5.2 Dokumentaatio

Prosessista koostettiin tarvittava dokumentaatio hyödyntäen Blue Prismin tarjoamia dokumenttipohjia. Prosessista koostettiin PDD- ja SDD-dokumentit. Dokumenttien perusrunko tehtiin ennen toteutuksen aloitusta ja niitä päivitettiin työn edistyessä. Dokumentaatio on tärkeässä roolissa prosessin vaatimuksen määrittämisessä sekä ylläpidossa. Opinnäytetyöhön liittyvät dokumentit PDD (26 sivua) ja SDD (25 sivua) ovat luokiteltu salaisiksi, eivätkä ole työn liitteenä.

PDD eli Process Definition Document on dokumentti, johon kuvataan liiketoiminta-prosessin työvaiheet sekä tekniset tiedot yksityiskohtaisesti. Dokumentissa on esitelty prosessin kulku, käytettävät kohdejärjestelmät, prosessin käyttämät henkilöresurssit sekä muita prosessiin liittyviä yksityiskohtia. Dokumenttiin on kuvattu yksityis-kohtaisesti prosessin kulku vaiheittain kuvankaappausten avulla. Dokumentti on koostettu yhdessä prosessiasiantuntijoiden kanssa.

SDD eli Solution Definition Document on dokumentti, jossa kuvataan robotiikka-ratkaisun tekninen toteutus. Dokumenttiin on eritelty robotiikkaratkaisun kulku sekä eri tehtävätyyppien keskeiset asiat. Esimerkiksi jokaisen tehtävätyypin lähettämät määrämuotoiset sähköpostit ja niistä poimittavat tiedot on eritelty tehtäväkohtaisesti. Lisäksi on eritelty kaikki mahdolliset poikkeustapaukset ja robotin toimintatapa niitä kohdattaessa. Dokumentissa on myös kerrottu, mitä Blue Prismin objekteja, prosesseja, ympäristömuuttujia, työjonoja ja käyttäjätunnuksia ratkaisussa käytetään. Dokumenttiin tulee myös kirjata mahdolliset työasemalle asetetut erityisvaatimukset tai asetukset.

5.3 Mallinnus

Mallinnus on työmäärällisesti projektin suurin työvaihe. Se koostuu työnkulun mallinnuksesta prosessitasolla sekä kohdejärjestelmien toiminnallisuuksien mallintamisesta objektitasolla. Tässä luvussa käydään läpi prosessin eri vaiheissa käytettyjä mallinnustekniikoita. Mallinnus on hyvä aloittaa kartoittamalla olemassa olevaa objektikirjastoa, jotta mahdolliset uudelleen käytettävät objektit tulevat hyödynnetyksi. Tämän robotisoinnin osalta kyseisiin kohdejärjestelmiin ei ollut valmiina objekteja, joten kaikkien objektien tekeminen oli osa projektia.

Mallinnuksen kanssa apuja löytyy Blue Prismin tarjoamasta portaalista https://portal.blueprism.com/. Sieltä löytyy ohjeistusta eri mallinnustekniikoiden käyttöön, sekä valmiit luurankopohjat objekteille ja prosesseille. Esimerkiksi prosessitason mallipohja ohjeistaa kehittäjää oikean prosessirakenteen ja poikkeushallinnan luomisessa. Nämä ovat tärkeitä asioita prosessin robustisuuden kannalta.

5.3.1 Tehtävien vastaanotto

Tehtävät vastaanotetaan sähköpostista, jonka käsittely on toteutettu hyödyntäen Blue Prismin mAPIEX-lisäosaa. Sen avulla on mahdollista käsitellä Outlookin sisältöä kooditasolla ilman varsinaista käyttöliittymää. Tässä prosessissa lisäosan avulla vastaanotetaan, lähetetään ja siirretään viestejä. Tehtäviä vastaanotettaessa luetaan Outlookin Saapuneet-kansiosta vanhin viesti ja tarkistetaan viestin lähettäjä ja otsikko. Mikäli viesti on tullut väärästä osoitteesta tai väärällä otsikolla, se jätetään käsittelemättä ja siirretään roskapostikansioon. Tiedot läpäisseet tehtävät otetaan käsiteltäväksi ja siirretään Outlookin sisällä keskeneräisten tehtävien kansioon. Viestin sisältö on määrämuotoinen ja sisältää tehtävän käsittelyyn tarvittavat tiedot. Seuraavassa esimerkissä on havainnollistettu viestin tyyliä kuvitteellisella viestimuotoilulla.

Tilaustiedot:
Tilausnumero: 123456789
Käyttäjätunnus: Q54321
Etunimi: Timo
Sukunimi: Toimihenkilö
Sähköposti: timo.toimihenkilo@esimerkki.fi
Esimies: Erkki Esimies (Q12345)
Valtuuden nimi: (PUH) PUHELINLUETTELO-sovelluksen HAKU-valtuus
Valtuuden tunnus: TUN4128821
Voimassaolon alkupäivä: 15.11.2016
Voimassaolon loppupäivä: 31.12.9999

Mikäli tekstistä tarvitaan esimerkiksi käyttäjän nimi, voidaan se eritellä viestitekstistä Calc-lohkon avulla. Esimerkiksi mikäli äskeinen esimerkkiviesti on tallennettu merkkijonona Data itemiin nimeltä "teksti", henkilön etunimi saadaan eriteltyä, kun Calc-lohkon sisältö on:

```
Trim(Mid([teksti]; (InStr([teksti]; "Etunimi:")+9); (InStr([teksti];
"Sukunimi:")-InStr([teksti]; "Etunimi:")-9)))
```

Visual Basicin InStr-komennon avulla määritetään merkkijonon sijainti toisessa merkkijonossa. Komento on muodossa InStr({Text}; {Search}), jossa Text on merkkijono, josta etsitään, ja Search merkkojono, jota etsitään. Komento palauttaa luvun, joka kertoo, monennestako merkistä kyseinen sana alkaa. Mid-komennolla voidaan erottaa merkkijonon keskeltä haluttu merkkijono. Komento on muodossa Mid({Text}; {Start point}; {Length}), jossa Text on alkuperäinen merkkijono ja Start point aloituspisteen luku ja Lenght halutun merkkijonon pituus. Trim -komennolla poistetaan välilyönnit ja rivinvaihdot merkkijonon alusta ja lopusta.

Vastaavalla tavalla poimitaan tekstistä kaikki tarvittavat tiedot. Kriittisten tietojen oikeellisuus on hyvä tarkistaa silloin, kun se on mahdollista. Jos esimerkissä olevan tunnus TUN4128821 olisi aina muotoa kolme kirjainta ja seitsemän numeroa, voitaisiin sille suorittaan tarkistus Decision-lohkon avulla seuraavasti:

```
Len([Valtuuden tunnus])=10 AND IsNumber(Right([Valtuuden tunnus]; 7))
```

Tällöin tarkastetaan, että merkkijonon pituus on 10 merkkiä ja että seitsemän viimeistä merkkiä ovat numeroita. Kun kaikki tiedot läpäisevät tarkistuksen, voidaan ne ladata Blue Prismin työjonoon odottamaan käsittelyä. Osassa tehtävistä puuttuu tehtävän kannalta välttämätön valtuusjärjestelmän käyttäjätunnus, joka haetaan toisen järjestelmän tietokannasta. Robotti käynnistää tämän järjestelmän ennen sähköposti-viestien käsittelyä, käyttää sitä tarvittaessa ja kun kaikki viestit on käsitelty, kirjautuu

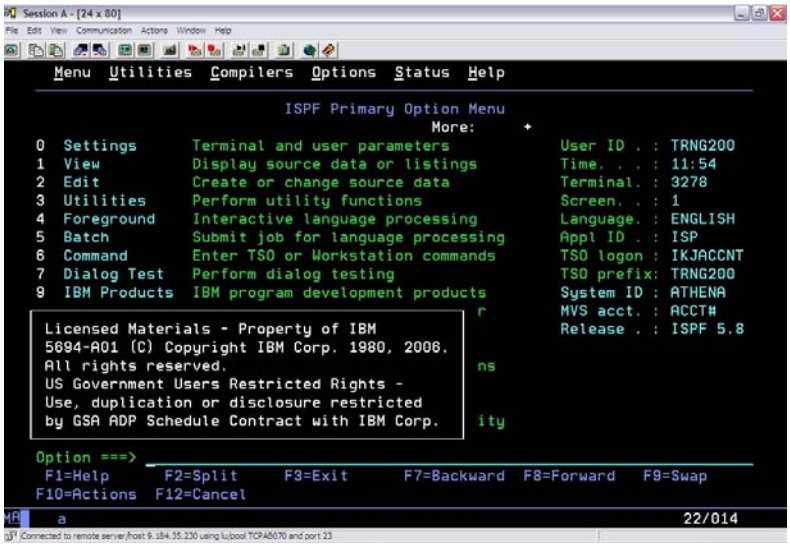
ulos järjestelmästä. Mikäli kaikkia kriittisiä tietoja ei löydetä, siirretään tehtävä sähköpostin avulla tiketöintijärjestelmään manuaalikäsiteltäväksi.

Blue Prismin työjonoon lisätään seuraavat tiedot:

- Koko sähköpostiviesti
- Sähköpostiviestin otsikko
- Sähköpostiviestin ID
- Käyttäjän etunimi
- Käyttäjän sukunimi
- Käyttäjän sähköpostiosoite
- Käyttäjän käyttäjätunnus
- Käyttäjän tunnus valtuusjärjestelmässä
- Tilaajan sähköpostiosoite
- Tehtävätyyppi (Tunnuksen luonti, valtuuden lisäys tai valtuuden poisto)
- Sovelluksen lyhenne
- Sovelluksen nimi
- Valtuuden tekninen nimi
- Valtuuden kuvausteksti

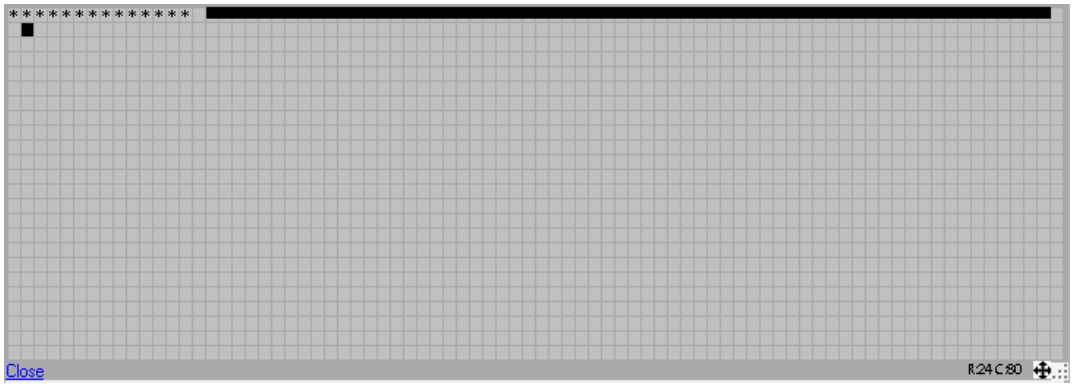
5.3.2 Tehtävien käsittely kohdejärjestelmässä

Kun tehtävät on saatu ladattua Blue Prismin työjonoon, siirrytään työvaiheeseen, jossa työjonosta käsitellään tehtäviä. Varsinainen käyttöoikeushallinta tapahtuu vanhassa IBM:n Mainframe Z/OS-käyttöjärjestelmässä. Mainframe-järjestelmissä laskenta on keskitetty suurtietokoneille. Järjestelmän käyttöliittymä on merkkipohjainen ja sen valikkorakenteissa liikutaan näppäimistön ja syötettävien komentojen avulla. Kuvassa 9 on esimerkki Z/OS-käyttöjärjestelmän näkymästä.



KUVA 9. IBM Mainframe Z/OS -käyttöjärjestelmä

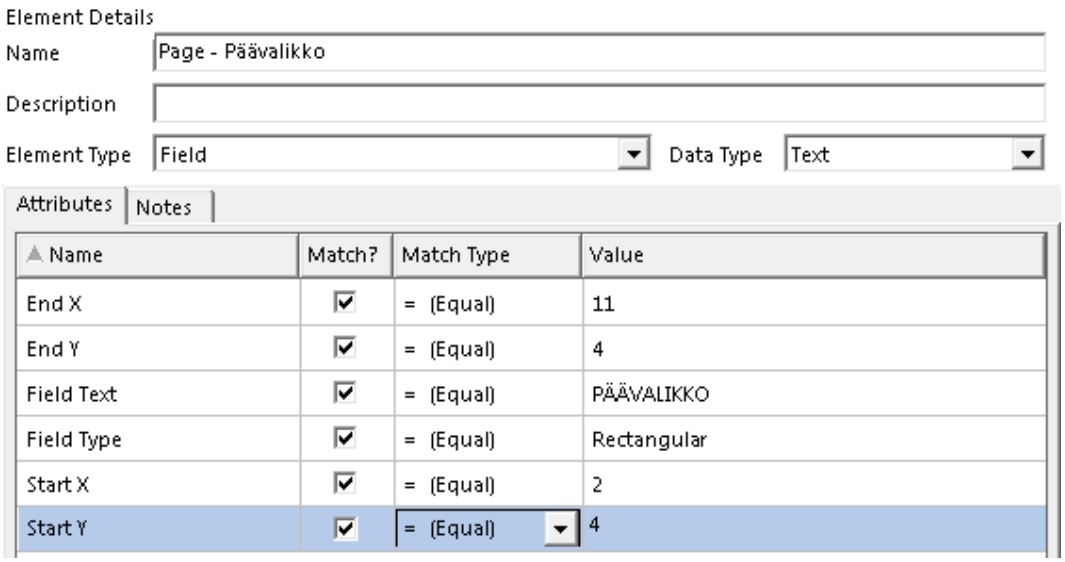
Järjestelmän käyttöliittymä on jaettavissa ruutuihin, joihin kuhunkin mahtuu yksi merkki. Kuvassa 10 näkyy Blue Prismin käyttöliittymän päälle muodostama ruudukko. Ruudukon koko on 24x80, mikä tarkoittaa 24 riviä, joille kullekin mahtuu 80 merkkiä.



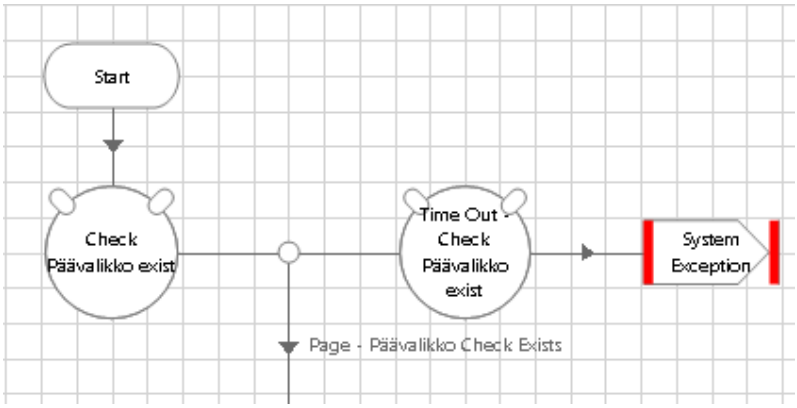
KUVA 10. Kohdejärjestelmän ruutunäkymä

Aluksi ohjelmistorobotti käynnistää kohdejärjestelmän ja kirjautuu sisään omilla tunnuksillaan. Mikäli robotin käyttämä salasana on vanhenemassa, suoritetaan salasanan vaihto generoimalla sattumanvarainen salasana, vaihtamalla se kohdejärjestelmässä sekä tallentamalla se Blue Prismin salattuun tietokantaan. Kun kohdejärjestelmään on kirjauduttu sisään onnistuneesti, navigoidaan valtuus-järjestelmään.

Kaikkien valikkonavigointien jälkeen ohjelmistorobotti jää odottamaan seuraavan ikkunan aukeamista. Mainframe eli merkkipohjaisissa sovelluksissa tarkistus tehdään odottamalla halutun tekstin ilmestymistä sille määritettyihin ruutukoordinaatteihin. Esimerkiksi mikäli päävalikko on tunnistettavissa 4 rivin 2 merkistä alkavasta "PÄÄVALIKKO" sanasta, voidaan se tunnistaa kuvan 11 mukaisesti. Tällöin päävalikkoon navigoitaessa jäädään odottamaan kyseisen tekstin ilmestymistä ennen kuin siirrytään seuraavaan toimenpiteeseen. Kuvassa 12 objektin toiminta aloitetaan tarkistamalla, että päävalikko on auki. "Check Päävalikko exist" -odotuslohkon sisälle on asetettu kuvan 13 mukainen parametri, jolla tarkistetaan kuvassa 11 määritetyn tekstin olemassaoloa määritetyissä koordinaateissa. Vasta kun teksti löytyy, jatketaan muiden toimintojen suorittamista. Mikäli oikea ikkuna ei ilmesty näkyviin määritetyn maksimiajan puitteissa, siirtyy prosessi poikkeustilaan.



KUVA 11. Application modeller -työkalun sisältö



KUVA 12. Wait-lohko


Actions

Element	Params	Condition	Type	Comparison	Value
Page - Päävalikko		Check Exists	Flag	= (Equal)	True

KUVA 13. Wait-lohkon sisältö

Samoja toimintoja hyödynnetään myös esimerkiksi valtuuden lisäyksen jälkeen. Silloin odotetaan kuittausta, jossa ilmoitetaan että valtuus on lisätty onnistuneesti. Kuitenkin on myös mahdollista saada esimerkiksi kuittaukset, että valtuus on jo voimassa tai että sitä ei voida lisätä. Tällöin robotille tulee määrittää toimenpiteet kaikissa näissä tilanteissa.

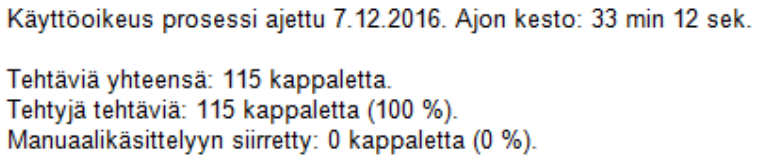
Tehtäviä poimitaan työjonosta yksi kerrallaan ja suoritetaan tehtävän vaatimat toimenpiteet. Halutut valtuudet kohdennetaan hakemuksesta työjonoon poimittujen tietojen perusteella. Prosessi jatkaa suoritusta, kunnes työjono on tyhjä tai tapahtuu prosessin kannalta kriittinen virhe. Lisäksi prosessi voidaan pysäyttää ylläpitokoneella olevasta valvontatyökalusta tai asettaa pysähtymään tiettynä kellonaikana.

5.3.3 Lähettävät viestit

Viestien lähetys on toteutettu hyödyntäen Blue Prismin mAPIEX-liitännäistä. Ohjelmistorobotti lähettää viestejä seuraavissa tilanteissa:

- Tehtävä on suoritettu onnistuneesti
  - Viesti palvelupyynnön käsittelystä valtuuskäsittelyn tilaajalle ja saajalle
- Tehtävä ei ole oikean tyyppinen tai tehtävän suoritus epäonnistuu
  - Alkuperäisen viestin välitys palvelupyyntönä tiketöintijärjestelmään
- Prosessi on keskeytynyt kriittisen tai toistuvan virheen takia
  - Vikaraportti prosessia valvovalle taholle
- Prosessi on suoritettu onnistuneesti loppuun
  - Ajoraportti prosessia valvovalle taholle

Sähköpostilla lähetettävä ajoraportti voidaan käsitellä esimerkiksi kuvassa 14 näkyvään muotoon. Mikäli ajossa on ilmennyt poikkeustapauksia, niistä koostetaan Excel-taulukko, joka lähetetään viestin liitteenä.



KUVA 14. Esimerkki robotin lähettämästä ajoraportista

Tällöin viestin teksti on muodostettu ajon aikana kerätyistä tiedoista (aloitusaika, onnistuneet tehtävät ja siirretyt tehtävät) ja muotoiltu Blue Prismin Calc-lohkolla seuraavasti:

```
"Käyttöoikeus prosessi ajettu " & Today() & ". Ajon kesto: "
&DateDiff(6; [starttime]; Now()) & " min " & DateDiff(3; [starttime];
Now())-(DateDiff(6; [starttime]; Now())*60)&" sek." & Chr(10) &
Chr(10) & "Tehtäviä yhteensä: " & [Count completed]+[Count exceptions]
& " kappaletta." & Chr(10) & "Tehtyjä tehtäviä: " & [Count completed]
& " kappaletta ("& Round((([Count completed] / ([Count exceptions]
+[Count completed]))*100); 1) & " %)." & Chr(10) &
"Manuaalikäsittelyyn siirretty: " & [Count exceptions] & " kappaletta
(" & 100-Round((([Count completed] / ([Count exceptions]+[Count
completed]))*100); 1) & " %)."
```

5.3.4 Poikkeustapausten hallinta

Prosessissa esiintyvistä virheistä käytetään Blue Prismissa nimityksiä "System exception" ja "Business exception". System exceptionilla tarkoitetaan järjestelmään liittyvää poikkeustapausta, joka voi muodostua esimerkiksi jonkin ikkunan käynnistyksessä ylittyvästä Time out -ajasta. Business exception on poikkeustapaus, joka liittyy liiketoimintaprosessiin. Esimerkiksi valtuuskäsittelyssä tällainen tilanne voi olla puuttuva tunnus tai valtuus, jota ei pystytä kohdentamaan. Kyseisissä tilanteissa on tarkoituksenmukaista, ettei robotti enää jatka kyseisen tehtävän käsittelyä.

Prosessin poikkeustapaustenhallinta toteutettiin siten, että liiketoiminnallisissa poikkeustapauksissa tehtävä siirretään suoraan manuaalikäsittelyyn, minkä jälkeen ohjelmistorobotti jatkaa seuraavaan tehtävään. Järjestelmävirheiden kohdalla tehtävä lisätään takaisin työjonoon. Järjestelmävirheen jälkeen kohdejärjestelmä suljetaan ja käynnistetään uudestaan, minkä jälkeen tehtävää yritetään uudelleen. Mikäli tehtävän kolmas käsittely-yritys epäonnistuu, ohjataan se manuaalikäsittelyyn. Tämän jälkeen työjonosta otetaan seuraava tehtävä. Mikäli seuraavassakin tehtävässä esiintyy sama

järjestelmävirhe, prosessin ajo pysäytetään ja tieto kriittisestä virheestä lähetetään prosessia valvovalle taholle.

5.4 Testaus

Testausta suoritettiin yhdessä mallinnustyön kanssa. Testaus suoritettiin kehitysympäristössä, johon oli luotu työtä varten testikäyttäjiä. Objektien testausta suoritettiin niiden mallinnuksen yhteydessä. Työn edistyttyä testausta laajennettiin prosessitasolle, jossa työn kulkua voitiin testata alusta loppuun.

Prosessille suoritettiin UAT eli User Acceptance Test, jossa robotin toiminta käytiin läpi sitä varten luodulla testiaineistolla. UAT suoritettiin yhdessä prosessi-asiantuntijoiden kanssa. Testiaineistoon pyrittiin koostamaan kaikki mahdolliset poikkeustapaukset, joita prosessissa saattoi ilmetä. Ensimmäisessä UAT:ssa havaittiin puutteita, jotka korjattiin ennen uusintatestausta. Uusintatestauksessa prosessi sai hyväksynnän tuotantoon siirrolle.

5.5 Tuotantoon siirto

Prosessin tuotantoon siirto suoritettiin joulukuussa 2016. Tuotantoon siirron jälkeen prosessille suoritettiin tuotantotestaus, jossa varmistettiin, että tehtävien suoritus onnistuu. Kun mallinnustyö tehdään testi- tai kehitysympäristössä, saattaa tuotantoympäristöön siirryttäessä ilmentyä eroavaisuuksia, jotka voivat estää robotin toiminnan. Tämän takia toiminta varmistetaan ajamalla robottia aluksi manuaali-käynnisteisesti. Tämän työn kohdalla havaittiin muutama eroavaisuus sovelluksen tuotanto- ja kehitysversioiden välillä. Lisäksi testauksen yhteydessä kohdattiin uusia liiketoiminnallisia poikkeustapauksia, joita ei osattu aiemmin huomioida. Prosessiin tehtiin tarvittavat muutokset, jotta ongelmat saatiin korjattua. Tuotantotestausta suoritettiin aluksi rajatulla määrällä valtuuksia ja niiden määrää kasvatettiin kun toiminta oli saatu todennettua.

Tuotantotestauksen valmistuttua joulukuun 2016 puolessa välissä, prosessi siirrettiin aikataulutukseen. Tällöin prosessi siirtyy ajettavaksi Runtime Resourcella, jolloin

robotti kirjautuu itse tietokoneelle ja käynnistää prosessin ajon. Kun ajo on valmis, kirjautuu robotti ulos koneelta. Aikataulutettuja käynnistyksiä voidaan määrittää prosessikohtaisten tarpeiden mukaan, esimerkiksi jokaisena pankkipäivänä tiettyyn kellonaikaan. Työn prosessin kohdalla tilaaja toivoi mahdollisimman monta ajoa arkipäiville, jolloin käsittelyaika saataisiin mahdollisimman lyhyeksi.

6 POHDINTA

Työn toteutus onnistui odotetulla tavalla, ja työn tilaaja oli tyytyväinen valmiiseen toteutukseen ja sen toteutusaikatauluun. Toteutus onnistui liiketoiminta-asiantuntijoiden kanssa ilman tarvetta ulkopuoliselle avulle, jonka myötä kustannukset pysyivät alhaisina. Nykyisellään prosessin investoinnin takaisinmaksuaika on 1–2 kuukautta, joka on erittäin hyvä tulos.

Joulukuussa 2016 robotti pystyi hoitamaan noin 80–90 % sille ohjatuista tehtävistä. Poikkeustapausten päivitysten myötä automaatioaste tulee asettumaan pidemmällä aikavälillä n. 90 % tietämille. Jäljelle jäävä 10 % osuus koostuu pääasiassa tehtävistä, joiden tekeminen ei ole mahdollista käsittelyhetkellä puuttuvien tietojen takia. On myös mahdollista, että robottia jatkokehitetään kattamaan myös muita valtuuskäsittelyn työtehtäviä.

Työ toi esille ohjelmistorobotiikan edut osana prosessikehitystä sekä mahdollisuudet puutteellisten sovellusintegraatioiden korvaajana. Tavallisella sovelluskehityksellä olisi vastaavassa automatisoinnissa kestänyt huomattavasti kauemmin sekä implementoinnin kustannukset olisivat olleet suuremmat.

Robotiikka ja tekoäly muokkaavat jo tänä päivänä työskentelytapojamme, ja onkin selvää että tietotyön automatisointisoinnilla tulee olemaan vaikutus työtilanteeseemme. Tämänkin prosessin automatisoinnin avulla tehtävien käsittely siirrettiin pääosin ulkoistuksesta takaisin yrityksen sisälle.

LÄHTEET

Barton, S. 2016. The state of Robotic Process Automation and Artificial Intelligence in the Enterprice. RPA & Artificial Intelligense Summit 2016. http://www.rpaandaisummit.com/rpa-and-ai-analysis-the-finance-industry-mloc-f-ty-m

Blue Prism. 2015. Blue Prism Infrastructure Overview Data Sheet (v4.2).

Business Wire. 2016. Software Robots Pioneer Blue Prism Debuts on the London Stock Exchange's AIM Market. Luettu 10.10.2016. http://www.businesswire.com/news/home/20160315005923/en/Software-Robots-Pioneer-Blue-Prism-Debuts-London

Kolehmainen, A. 2015. Robotit mullistavat työnteon. Tivi, Joulukuu 2015. Talentum.

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

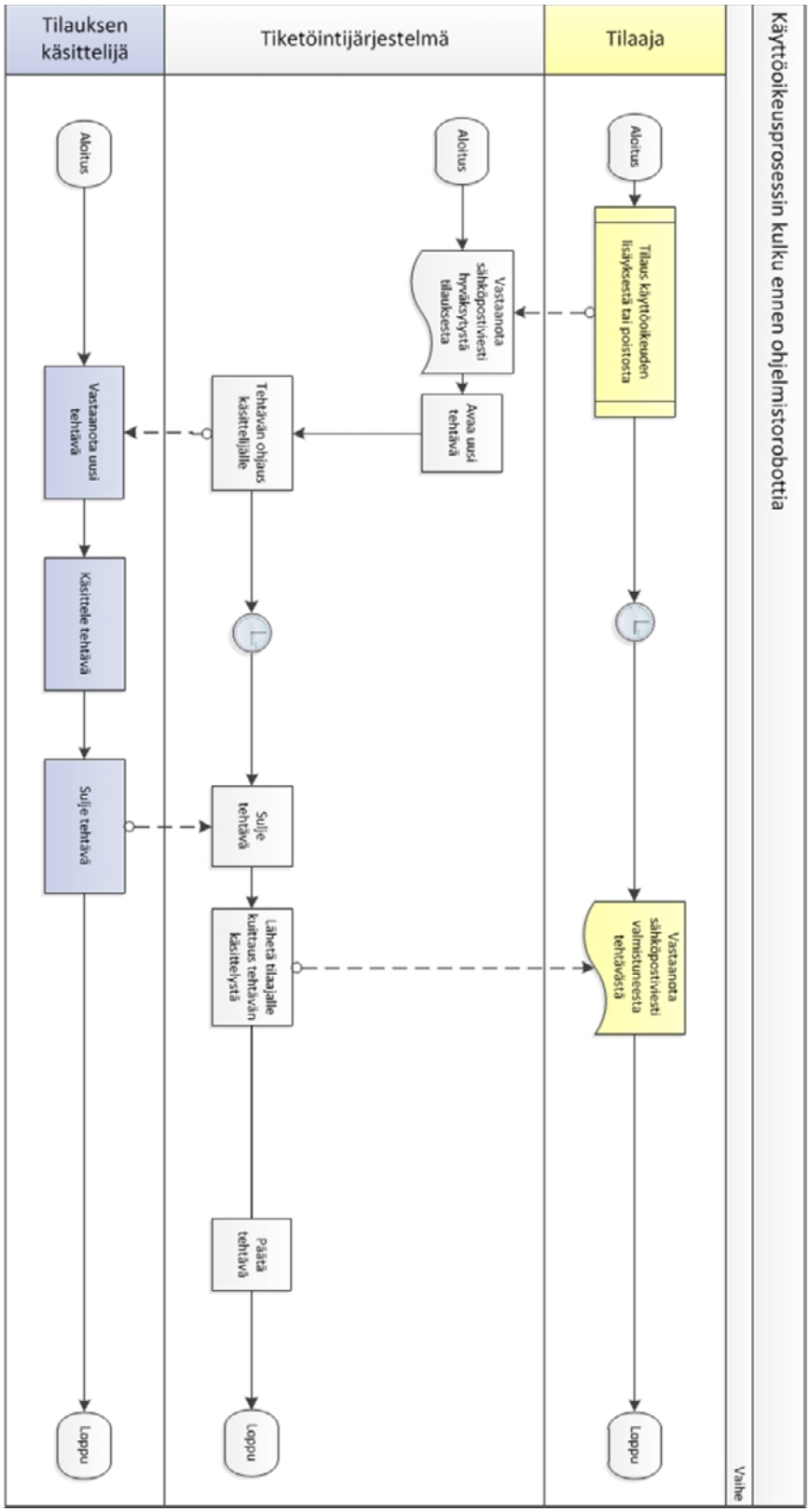
Sajari, P. 2015. Taloustieteilijä Holmström: Suomessa ei ole ymmärretty, kuinka huonossa kunnossa talous on. HS 19.8.2015. Luettu 20.10.2016. http://www.hs.fi/talous/art-2000002845992.html

Tamminen, O. 2016. Työelämää mullistava ohjelmistorobotti uurastaa väsymättä. Fujitsu Net-lehti 1/2016. http://net.fujitsu.fi/fi-FI/12016/Tyoelamaa_mullistava_ohjelmistorobotti_u(9656)

Willcocks, P. & Lacity, M. 2016. Robots and The Future of Work. United Kingdom: Steve Brookes Publishing.

LIITTEET

Liite 1. Pelkistetty prosessikaavio ennen robotisointia



Liite 2. Pelkistetty prosessikaavio robotisoinnin jälkeen