PROSESSIMALLINNUKSEN JA -AUTOMAATION SYNERGIA TOIMIHENKILÖTYÖN SUORITUSKYVYN PARANTAMISESSA

Lappeenrannan–Lahden teknillinen yliopisto LUT

Tuotantotalouden diplomityö

2023

Vilma Valtamo

Tarkastajat: Dosentti Jouni Koivuniemi

Tutkijatohtori Lasse Metso

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tuotantotalous

Vilma Valtamo

Prosessimallinnuksen ja -automaation synergia toimihenkilötyön suorituskyvyn parantamisessa

Tuotantotalouden diplomityö

2023

113 sivua, 26 kuvaa, 12 taulukkoa ja 14 liitettä

Tarkastajat: Dosentti Jouni Koivuniemi ja tutkijatohtori Lasse Metso

Avainsanat: Prosessijohtaminen, prosessimallinnus, prosessiautomaatio, ohjelmistorobotiikka, toimihenkilötyön tehostaminen

Työn tarkoituksena oli selvittää prosessimallinnuksen ja -automaation yhdessä tuomia hyötyjä tarjous-hankintaprosessin toimihenkilötyön tehostuksessa. Tiukentuvan markkinatilanteen vuoksi yritysten on löydettävä uusia tapoja toimintansa tehostamiseksi, joten moni yritys on kiinnostunut erilaisista toimihenkilötyön tehostamisen keinoista. Työn tavoitteena oli mallintaa ANDRITZ Savonlinna Works Oy:n tarjous-hankintaprosessi sen nykyisessä tilassa, paikantaa siihen perustuen mahdollisia automaatiokohteita ja toteuttaa prosessin osan suorituskykyä parantava automaatiodemo. Työlle luotiin ensin teoreettinen pohja, jonka jälkeen toteutettiin haastattelukierrokset, joiden pohjalta rakennettiin mallinnus käyttäen BPMN-syntaksia. Automatisoitava prosessin osa valittiin mallinnuksesta ja automaatio toteutettiin Python-ohjelmointikielellä. Tuloksena syntyi todenmukainen mallinnus yrityksen tarjous-hankintaprosessista sekä siinä liikkuvasta tiedosta. Tämän pohjalta rakennettiin automaatiodemo, joka koostuu kolmesta ohjelmasta, joista ensimmäinen luo, muotoilee ja täyttää kolme tiedostoa perustuen syötettyyn ostojonoon ja jos tietoja puuttuu, kutsutaan toista ohjelmaa, joka hakee puuttuvat tiedot hankintasuunnitelmasta. Kolmas ohjelma on ohjelmistorobotti, joka syöttää luodut tiedostot oikeille paikoilleen. Ratkaisun arvioitiin parantavan prosessin osan laadullista, ajallista ja taloudellista suorituskykyä. Työhön perustuvien johtopäätösten mukaan prosessimallinnus ja -automaatio tukevat toisiaan useilla tavoilla. Automatisoitavan prosessin osan löytäminen helpottui huomattavasti, samoin kuin itse automaation toteutus, kun prosessin tiedonkulku ja prosessilogiikka olivat selvillä. Prosessimallinnus voi myös tukea automaation ylläpitoa siinä tapauksessa, että mallia pidetään systemaattisesti ajan tasalla. Pidemmän aikavälin hyötyjä on jäätävä seuraamaan.

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

School of Engineering Science

Industrial Engineering and Management

Vilma Valtamo

The synergy between Business Process Modeling and Business Process Automation in improving the performance of white-collar work

Master's thesis

2023

113 pages, 26 figures, 12 tables and 14 appendices

Examiners: Docent Jouni Koivuniemi and post-doctoral researcher Lasse Metso

Keywords: Business Process Management, Business Process Modeling, Business Process Automation, Robotic Process Automation, white-collar work performance

The purpose of this thesis work was to investigate the synergy between Business Process Modeling and Business Process Automation in improving the performance of white-collar work in the quotation-to-acquisition part of the business' model. The increasingly challenging and competitive market environment has forced businesses to find new ways of improving their performance, which has led to increasing interest in the streamlining of white-collar work. The objective of this thesis work was to model ANDRITZ Savonlinna Works Oy's quotation-to-acquisition process as it is now, finding potential automation targets through that modeling, and to build a demonstration of automation for that process. First, a theoretical framework was established for this thesis, after which a series of interviews were held. Based on those a complete model was build using a BPMN syntax. After a suitable part for automation was chosen from the model, the automation work was done utilizing the Python programming language. The result was an accurate model of the quotation-to-acquisition process and its flow of information. Based on the model three programs were developed. The first one creates, formats, and fills in the information in three separate Excel tables based on the purchasing queue. Any missing information is fetched by the second program, which searches for it in the acquisition plan. The third program is a software robot, which imports the three tables created in the first program. It was estimated that the resulting automation solution would better the processes' qualitative, temporal, and economic performance. Based on the conclusions drawn from this thesis, Business Process Modeling and Automation aid each other in many ways. Finding a suitable part for automation and developing the automation were made significantly easier by the existence of the full model. If the Business Process Model is kept systematically up to date, it could also help in the upkeeping of the automation. Any long-term indicators need to be monitored and followed up on in the future.

Sisällysluettelo

Tiivistelmä

Abstract

Alkusanat

1 Johdanto ... 8
1.1 Työn tausta, tutkimuskysymykset ja rajoitteet ... 8
1.2 Työn menetelmät ... 9
1.3 Tutkimuksen rakenne ... 11
2 Prosessijohtaminen ja -mallinnus ... 14
2.1 Prosessijohtamisen elinkaaren vaiheet ... 15
2.2 Prosessin suorituskyvyn mittaaminen ... 18
2.3 Toimihenkilötyön määritelmä ja mittaus ... 21
2.4 Liiketoimintaprosessien mallintaminen ... 22
2.5 Prosessimallinnuksen erilaiset näkökulmat ... 24
2.6 Mallin kuvaustavan ja notaation valinta ... 25
3 Prosessiautomaatio ... 28
3.1 Liiketoimintaprosessien automaatio ... 28
3.2 Ohjelmistorobotiikan kuvaus ... 29
3.3 Automatisoitavan prosessin valintakriteerien asetus ... 30
3.4 Automaatioprojektin toteutus ... 32
4 Tutkimusprosessin kulku ... 36
4.1 ANDRITZ Savonlinna Works Oy:n esittely ... 36
4.2 Tutkimusprosessin kulku yleisellä tasolla ... 37
4.3 Haastattelut mallinnuksen pohjana ... 39
4.4 Mallinnusprosessin eteneminen vaihe kerrallaan ... 41
4.5 Tiedonkulku ja ohjelmistot mallinnuksessa ... 48
4.6 Automatisoitavan prosessin osan valinta ... 50
4.6.1 Hankinnan alkusuunnittelun nimiketietojen hakeminen tietokannasta ... 50
4.6.2 Hankinnan suunnittelun Excel-osaluettelon valmistelu ... 51
4.6.3 Hankinnan suunnittelun viimeiset SAP-merkinnät ... 52
4.6.4 Puuttuvien tietojen korjaaminen automaattisen oston mahdollistamiseksi ... 53

4.7 Automaatioratkaisun toteutus 55
5 Työn ratkaisujen esittely 60
5.1 Toteutetun mallinnuksen tulokset 60
5.2 Automaatiodemon toiminta 63
5.2.1 Luo Excel-tiedostot 65
5.2.2 Hae tietoa hankintasuunnitelmasta 70
5.2.3 Ohjelmistorobotti SAP-syöttöihin 75
5.3 Suorituskyvyn muutoksen arviointi 80
6 Johtopäätökset 84
6.1 Vastaukset tutkimuskysymyksiin 84
6.2 Tulosten hyödynnettävyys ja rajoitteet 86
6.3 Jatkokehityskohteet 87
7 Yhteenveto 89
Lähdeluettelo 91

Liitteet

Liite 1. Prosessimallinnuksen haastattelukysymykset

Liite 2. Tarjous-myyntiprosessi

Liite 3. Projektin alkutoimenpiteet

Liite 4. Hankinnan alkusuunnittelu

Liite 5. Hankinnan suunnittelu

Liite 6. Operatiivinen osto

Liite 7. Tee auto-po-tilaukset

Liite 8. Korjaa puuttuvat tiedot

Liite 9. Pätkä Luo_Excelit.py ohjelmakoodista, josta luottamuksellinen tieto sekä käyttäjänimet on piilotettu

Liite 10. Pätkä Haku.py ohjelmakoodista, josta luottamuksellinen tieto sekä käyttäjänimet on piilotettu

Liite 11. Pätkä SAP_robotti.py ohjelmakoodista

Liite 12. Simuloitu SAP Production System -etusivu

Liite 13. Simuloitu ZMMINFOREC-transaktion sivu

Liite 14. Simuloitu Import purchasing Inforecord -sivu

Kuvaluettelo

Kuva 1. Työn teoreettinen viitekehys

Kuva 2. Käsitteiden väliset suhteet ja niihin vaikuttavat trendit

Kuva 3. Prosessijohtamisen elinkaari on jaksottainen prosessi

Kuva 4. Automaatioprojektin kulku ja eri vaiheissa huomioitavat asiat

Kuva 5. Diplomityön hierarkkisen mallinnuksen logiikka

Kuva 6. Mallin tehtävien kronologinen hahmotelma

Kuva 7. Malli järjestysvuolla ja logiikkaporteilla

Kuva 8. Malli tehtävien roolien, silmukoiden sekä viestivirtojen kanssa

Kuva 9. Osaprosessin linkkitapahtuman kumpikin puoli

Kuva 10. Kommenteilla viimeistelty osaprosessi

Kuva 11. Esimerkki ohjelmistojen ja tiedonkulun esittämisestä osana prosessia

Kuva 12. Tietovarastojen ja tiedonkulkua kuvaavien nuolten värikoodaus

Kuva 13. Automaattisen tilauksen tarvitsemat tiedot ja niiden lähteet

Kuva 14. Koko mallinnettu prosessi korkealta tasolta

Kuva 15. Automaatiodemon ohjelmien ja tiedonkulun mallinnus

Kuva 16. Inforecord-tiedoston luominen

Kuva 17. Plant-tason inforecord-tiedoston luominen

Kuva 18. Päivitetyn inforecord-tiedoston luominen

Kuva 19. Luo_Excelit.py tiivistettynä

Kuva 20. Hankintasuunnitelman rakenne

Kuva 21. Haku.py tiivistettynä

Kuva 22. Täytetty sisäkkäinen sanakirja tulostettuna (luottamuksellinen tieto peitetty)

Kuva 23. Inforecord-tiedoston syöttö

Kuva 24. Plant-tason inforecord-tiedoston syöttö

Kuva 25. Päivitetyn inforecord-tiedoston syöttö

Kuva 26. SAP_robotti.py tiivistettynä

Taulukkoluettelo

Taulukko 1. Diplomityön input/output -taulukko

Taulukko 2. Yhteenveto prosessin suorituskyvyn perusmittareista sekä niiden oleellisimmista osa-alueista

Taulukko 3. Prosessimallinnuksen näkökulmat ja ominaisuudet (per. Luukkonen et al. 2012)

Taulukko 4. Liiketoimintaan sekä automaatioon liittyvät valintakriteerit ja tekijät

Taulukko 5. Vertailu back-end- ja front-end-automaation välillä (per. Santos et al. 2020)

Taulukko 6. Tutkimusprosessin kulku

Taulukko 7. Palaverien kestot, osallistujat ja niistä saatu tieto

Taulukko 8. Vertailu siitä, miten hyvin eri vaihtoehdot täyttävät asetetut automaatiokriteerit

Taulukko 9. Tiedot, joita vaaditaan kussakin SAP:iin syötettävässä tiedostossa

Taulukko 10. SAP-robotissa käytettyjen bittikarttojen nimet ja kuvat

Taulukko 11. Ohjelmien suoritusten aikahaarukat

Taulukko 12. Vertailu käsin ja ohjelmallisesti syötettävien aikojen välillä

1 Johdanto

Digitalisaatio, kansainvälistyminen ja tiukentuva kilpailutilanne vaativat yrityksiä tehostamaan toimintaansa jatkuvasti. Erityisesti teknologia- ja teollisuusyrityksissä tämä on perinteisesti toteutettu paremmilla laitteilla ja laitoksilla, mutta näiden tehostustoimien ehtyessä yritykset ovat ruvenneet enemmissä määrin kiinnostumaan asiantuntijatyön tehostamisesta. Niin kutsuttu toimihenkilötyö on vaikeammin määriteltävissä, mitattavissa ja korvattavissa automaatiolla, mutta sen suorituskyvyn tehostamiseen löytyy nykyään paljon keinoja ja kiinnostusta. Nämä keinot liittyvät erityisesti prosessijohtamiseen sekä ohjelmalliseen tehostukseen, kuten automaatioon ja ohjelmistorobotteihin. (Ahmad & van Looy 2020; Ferraris et al. 2018; Ilmarinen & Koskela 2015, 51–69)

1.1 Työn tausta, tutkimuskysymykset ja rajoitteet

Tässä diplomityössä haluttiin tarkastella erityisesti prosessimallinnuksen ja -automaation yhteisvaikutusta toimihenkilötyön suorituskyvyn parantamisessa. Tämä työ tehtiin luottamuksellisesti ANDRITZ Savonlinna Works Oy:lle, joka on Savonlinnassa toimiva sellutehtaiden laitteita toimittava teknologia- ja palveluyritys. Yrityksessä työtä tukivat yritysohjaaja sekä erilaisten funktioiden edustajista koottu ohjausryhmä. Diplomityön tarkastelualue rajattiin yrityksen tarjous-hankintaprosessiin, sillä se on yrityksen koko liiketoimintaprosessin kannalta erittäin kriittinen vaihe. Työssä haluttiin myös välttää prosessin osia, jotka ovat kosketuksissa yrityksen toiminnanohjausjärjestelmän eli SAP-rajapinnan kanssa, sillä sopimusteknisistä syistä sen automaattinen käsittely on tällä hetkellä harmaalla alueella. Lisäksi yritys on tämän vuoden aikana siirtymässä uuteen SAP-versioon.

Näiden tavoitteiden ja rajausten tueksi koottiin yksi päätutkimuskysymys ja kolme apukysymystä.

Miten toimihenkilötyön suorituskykyä voidaan parantaa tarjous-hankintaprosessissa?

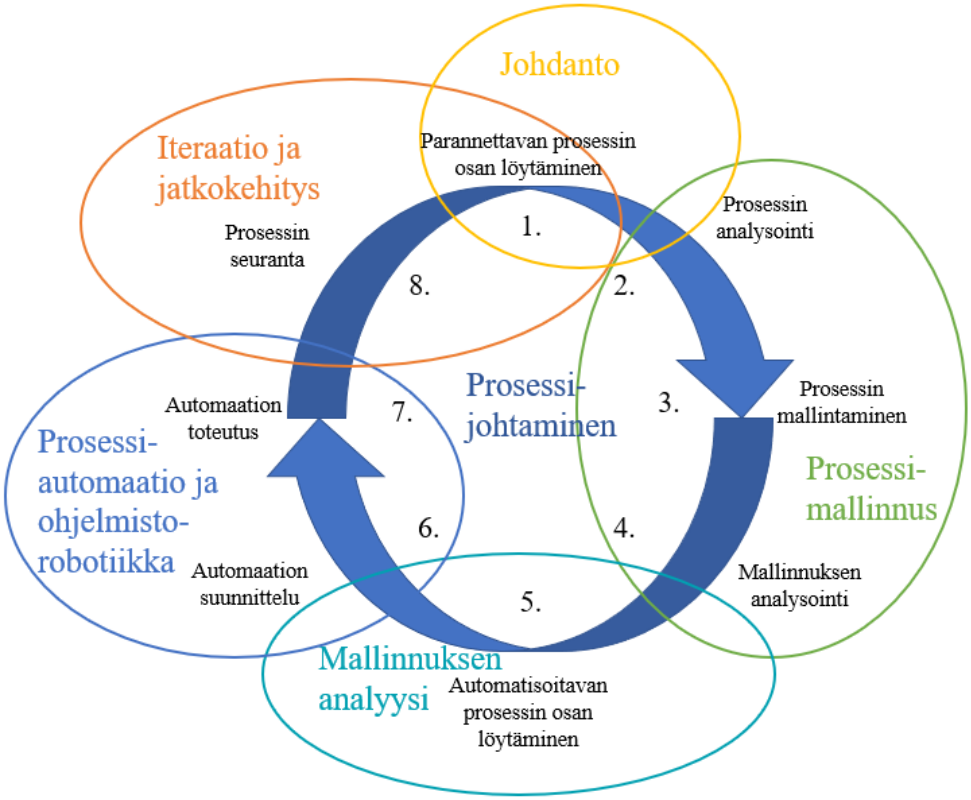
1. Miten prosessimallinnus voi tehostaa toimihenkilötyötä?
2. Miten prosessiautomaatio hyödyttää toimihenkilötyötä?

3. Miten prosessimallinnus ja -automaatio voivat tukea toisiaan prosessin suorituskyvyn parannuksessa?

Ensisijaisesti haluttiin selvittää, miten toimihenkilötyön suorituskykyä voidaan parantaa tarjous-hankintaprosessissa. Työssä valittujen menetelmien pohjalta apukysymyksiksi valittiin selvitys siitä, miten prosessimallinnus ja prosessiautomaatio kukin voivat vaikuttaa toimihenkilötyön tehostamiseen. Prosessimallinnuksen ja -automaation yhteisvaikutusta toimihenkilötyön suorituskykyyn haluttiin myös arvioida, joten siitä muodostettiin kolmas apukysymys. Suorituskykyä voidaan mitata ja arvioida monella tavalla, mutta yritys oli erityisen kiinnostunut toimihenkilöiden ajallisen, taloudellisen ja laadullisen suorituskyvyn tehostamisesta tarjous-hankintaprosessissa.

1.2 Työn menetelmät

Tutkimuksen taustoituksen ja rajoitteiden asettamisen jälkeen määritellään työn teoreettinen viitekehys. Prosessijohtamisen ja -mallinnuksen alaiset konseptit ja käsitteet tarkastellaan kuvailevassa kirjallisuuskatsauksessa integroivaan tyyliin, jonka jälkeen vastaava käsittely tehdään prosessiautomaatiolle. Näiden pohjalta työn teoreettinen viitekehys saadaan syntetisoitua malliksi, jossa prosessimallinnus- ja automaatio on yhdistetty samaan prosessijohtamisen elinkaareen perustuvaan jaksolliseen malliin. Kuvaan 1 on tiivistetty tämän mallin lisäksi työn tutkimustoteutuksen päävaiheet. Malli on jaksottainen esitys jonkin liiketoimintaprosessin osan parannuksesta ensin korjattavan prosessin osan tunnistuksella, sen mallinnuksella ja lopuksi sen automaatiolla. Mallinnus- ja automaatioprosesseja yhdistää mallinnuksen analyysi, jossa sopiva kohta automaatiolle löydetään mallinnuksen pohjalta. Iteraatio ja jatkokehitys aloittavat jakson uuden kierroksen. Mallin päälle on myös asetettu työn tutkimuksen päävaiheet suunnilleen niihin kohtiin, joihin ne mallissa liittyvät.



Kuva 1. Työn teoreettinen viitekehys ja tutkimustoteutuksen päävaiheet

Työn teoreettisen pohjan luomiseksi käytettiin vertaisarvioituja artikkeleita ja oppikirjoja, joiden löytämisessä hyödynnettiin pääasiassa lähdetietokantoja SCOPUS ja Primo. Tiedonhaussa hyödynnettiin avainsanoja kuten, ’business process management’, ’business process modeling’, ’robotic process automation’, ’software robotics’ ja ’white-collar productivity’. Prosessimallinnus toteutettiin haastattelemalla toimihenkilöitä jokaisesta oleellisesta osaprosessista ja yhdistämällä näiltä saatu tieto yrityksen prosessi- ja laatudokumentaatioon. Mallinnuksen, automaatiokriteeristön ja osaprosessien asiantuntijoiden kertomusten pohjalta löydettiin sopiva prosessin osa automaatiolle. Tämän pohjalta rakennettiin automaatiodemo ja analysoitiin sen tuomia etuja, kuten suorituskyvyn eri osa-alueiden parannusta. Automaatiodemossa hyödynnetään Pythonin runsasta moduulivalikoimaa sekä muita ohjeita ja dokumentteja niiden hyödyntämiseksi. Tällä tavalla saatiin konkreettinen kuva prosessimallinnuksen ja -automaation yhteistoiminnasta ja samalla pyrittiin tehostamaan kohdeyritykselle strategista prosessin osaa ja arvioimaan kehityskohteita yrityksen tulevia kehityskohteita varten. Tutkimuksen tarkempi toteutus määritellään myöhemmin luvussa 4. Työ toteutettiin ajanjaksolla 2.1.–24.5. vuonna 2023 Savonlinnassa.

1.3 Tutkimuksen rakenne

Työn tarkempaa rakennetta on kuvattu Taulukossa 1 input/output -kaavion kautta. Kaavion tarkoituksena on kuvata, mitä tietoa mikäkin luku tarvitsee ja mitä oleellista tietoa siitä saadaan ulos. Johdannossa hyödynnettiin tietolähteenä aiemmin laadittua diplomityön projektisuunnitelmaa sekä työn koko taustaa ja toteutusta, sillä johdanto kirjoitettiin viimeisenä. Tuloksena tästä luvusta saatiin työn tutkimuskysymykset sekä tausta ja perustelut työn toteutukselle. Työn kaksi ensimmäistä sisältölukua kuvaavat prosessijohtamisen ja -automaation teoriataustaa ja niihin liittyviä käsitteitä. Näistä koottu teoreettinen viitekehys on tiivistetty Kuvaan 1. Työn ratkaisujen kannalta oleellisimmat tulokset, jotka näistä luvuista saatiin, olivat työssä käytetty mallinuksen notaatiokieli, toimihenkilötyön mittaaminen, automaatioprojektin suunnittelu ja haasteet sekä monipuolisempi automaatiokriteeristö.

Taulukko 1. Diplomityön input/output -taulukko

Input	Otsikko	Output
• Työn tausta ja toteutus • Diplomityön projektisuunnitelma	Luku 1. Johdanto	• Työn tutkimuskysymykset • Työn taustoitus
• Kirjallisuus prosessijohtamisesta ja -mallinnuksesta • Työn tavoitteet	Luku 2. Prosessijohtaminen ja -mallinnus	• Prosessijohtamisen elinkaari • Toimihenkilötyön mittaaminen • Prosessimallinnuksen tarkoitus, tarpeet ja näkökulmat • Työssä käytettävä notaatiokieli
• Kirjallisuus prosessiautomaatiosta ja ohjelmistoroboteista • Toimihenkilötyön mittaaminen • Prosessijohtamisen elinkaari	Luku 3. Prosessiautomaatio	• Digitalisaation käsitteet • Liiketoimintaprosessien automaation käsitteet • Yleiset automaation kriteerit • Ohjelmistorobottien toiminta, hyödyt ja haasteet • Automaatioprojektien suunnittelu ja haasteet • Erot back-end ja front-end-automaation välillä • Monipuolisempi automaatiokriteeristö
• Työn tutkimuskysymykset • Asiantuntijoiden haastattelut • Työssä käytettävä notaatiokieli • Automaatioprojektien suunnittelu ja haasteet • Monipuolisempi automaatiokriteeristö	Luku 4. Tutkimusprosessin kulku	• Haastatteluiden rakenne • Mallien rakennuksen selitys • Automatisoitava prosessin osa • Automaatiodemon suunnitelma
• Toimihenkilötyön mittaaminen • Asiantuntijoiden haastattelut • Mallien rakennuksen selitys • Automatisoitava prosessin osa • Automaatiodemon suunnitelma	Luku 5. Työn ratkaisujen esittely	• Mallien esittely ja tulokset • Automaatiodemon esittely ja tulokset • Suorituskyvyn muutoksen arviointi
• Työn tutkimuskysymykset • Mallien esittely ja tulokset • Automaatiodemon esittely ja tulokset • Suorituskyvyn muutoksen arviointi	Luku 6. Johtopäätökset	• Analyysi työn tuloksista ja onnistumisesta • Jatkokehitysideoita ja ehdotuksia tulevaisuudelle
• Kaikkien aikaisempien lukujen tulokset	Luku 7. Yhteenveto	• Tiivistelmä koko työn teoriasta, menetelmistä ja tuloksista

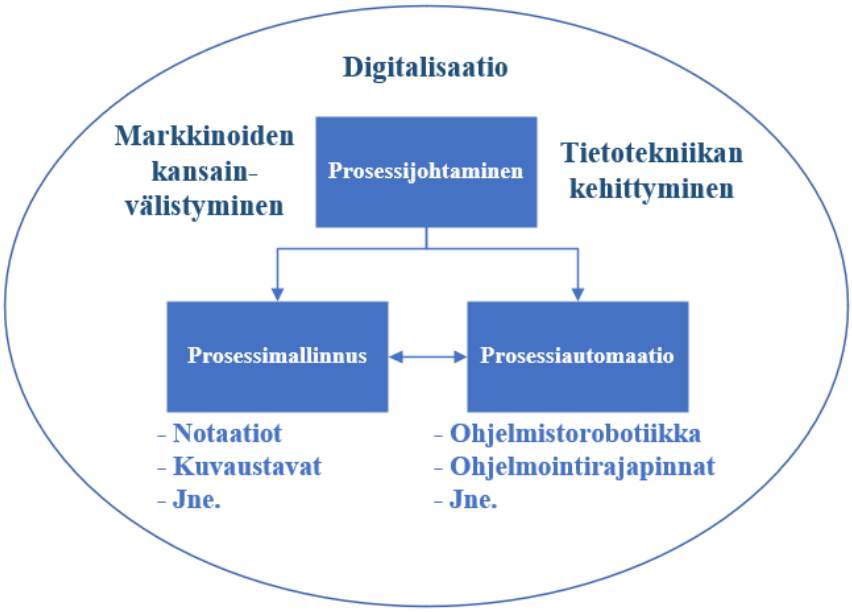
Teoreettisen viitekehyksen selvityksen jälkeen siirryttiin työn toteutettavaan vaiheeseen. Neljännessä luvussa esitellään tarkemmin tutkimusprosessin kulkua, kuten haastatteluiden rakennetta, mallinnuksen logiikkaa, automatisoitavan prosessin osan löytämistä sekä sen

pohjalta tehtyä automaatiodemon suunnitelmaa. Näiden pohjalta siirrytään viidenteen lukuun, jossa toteutettujen ratkaisujen toimintaa esitellään, niiden tulokset eritellään ja niiden vaikutuksia analysoidaan. Johtopäätöksissä vedetään kaikki tämä yhteen ja vastataan työn tutkimuskysymyksiin sekä pohditaan työn tuloksia. Lisäksi luvussa tehdään joitain jatkokehitysehdotuksia tulevaisuutta varten. Yhteenveto käyttää lähteenään kaikkia aikaisempia lukuja ja tiivistää ne yhdeksi luvuksi.

2 Prosessijohtaminen ja -mallinnus

Prosessijohtamisella (engl. Business Process Management, BPM) tarkoitetaan kaikkia niitä toimia, joiden avulla työn kulkua seurataan ja hallitaan, jotta työn tulos pysyisi mahdollisimman johdonmukaisena ja laadukkaana. Organisaation sisäisen liiketoimintaprosessin löytäminen, kuvaaminen, mittaaminen, kehittäminen ja optimointi ovat prosessijohtamisen oleellisimmat määränpäät. Liiketoimintaprosessilla tarkoitetaan sitä prosessien ketjua, joka johtaa tuotteen tai palvelun toimittamiseen asiakkaalle. Prosessijohtamisen voidaan katsoa koostuvan eritasoisista elementeistä, joista tärkeimmät ovat tapahtumat, aktiviteetit ja päätöksentekokohdat. Prosessijohtamiseen kuuluu aina tietty määrä joko sisäisiä tai ulkoisia toimijoita (actors) sekä esineitä (objects), joiden ei välttämättä tarvitse olla aineellisia. Johtamisprosessissa on myös aina jokin tai joitain tuotoksia, joka tai jotka toimitetaan lopuksi asiakkaalle, joka saa tuotoksesta itselleen jotain arvoa. (Dumas et al. 2013, 1–6; Mendling 2022; Peña et al. 2022)

Tämän diplomityön osalta on huomattava, että prosessimallinnus ja -automaatio (engl. Business Process Automation, BPA) sekä niiden alakäsitteet kuuluvat prosessijohtamisen piiriin. Prosessimallinnus on erityisen läheinen osa prosessijohtamista, sillä sen tarkoituksena on antaa totuudenmukainen kuva yrityksen toiminnoista ja muuttaa hiljainen tieto avoimeksi (Ahmad & van Looy 2020). Prosessiautomaatio taas liittyy vahvasti prosessien kehittämiseen ja optimointiin, joilla taas on vahva kytkös prosessijohtamiseen. Prosessiautomaatio ja -mallinnus esiintyvät myös usein yhdessä automaatioprojekteissa. (Dumas et al. 2013, 6; Paschek et al. 2017) Erityisesti digitalisaatio ja markkinoiden kansainvälistyminen ovat olleet voimakkaina ajureina prosessijohtamisen muutokselle, kun kiristynyt kilpailu vaatii yrityksiltä jatkuvasti tehokkaampaa ja reaktiokykyisempää toimintaa (Ahmad & van Looy 2020; Ferraris et al. 2018). Digitalisaatio on lisännyt yritysten kasvu- ja suorituskykypotentiaalia huomattavasti ja siten kiihdyttänyt kilpailua entisestään eteenkin teollisuuden aloilla (Ilmarinen & Koskela 2015, 51–69). Se miten kaikki tässä diplomityössä oleellisimmat käsitteet liittyvät suhteessa toisiinsa on kuvattu Kuvassa 2. Kuvasta myös näkee viimeaikaiset trendit, joilla on ollut vaikutusta näihin toimintoihin.



Kuva 2. Käsitteiden väliset suhteet ja niihin vaikuttavat trendit

Digitalisaatio ja tietotekniikan kehitys ovat olleet mahdollistamassa, mutta myös vauhdittamassa sekä prosessimallinnuksen että prosessiautomaation käyttöönottoa. Prosessimallinnusta käsitellään prosessijohtamisen alla niiden läheisen yhteyden takia ja prosessiautomaatiota sekä sen alaisuuteen kuuluvia konsepteja käsitellään niiden laajuudesta johtuen omassa luvussaan myöhemmin.

2.1 Prosessijohtamisen elinkaaren vaiheet

Prosessijohtamisen toteutuksessa käytetään perinteisesti niin kutsuttua BPM-elinkaarimallia. Mallia käytetään, koska yritysten on jatkuvasti seurattava, parannettava ja uudistettava toimintaansa pysyäkseen kilpailullisina. (Ruževičius et al. 2012) Dumas et al. (2013, 15) mukaan malliin kuuluu kuusi perusvaihetta, jotka esitellään seuraavaksi.

Ensimmäisessä vaiheessa on tunnistettava ongelma, johon halutaan ratkaisu ja ongelmaan liittyvät prosessit sekä tehtävät ja muu ratkaisun löytämiseen tarvittava tieto. Tehtävät tulisi myös hahmottaa osana laajempaa liiketoimintaprosessia, jotta tiedetään, mitä vaatimuksia

niiden tulisi täyttää. (Ruževičius et al. 2012) Tässä kohdassa on myös hyvä tehdä tarkat rajaukset siitä, mitkä prosessit kuuluvat ongelman piiriin. Tarkoituksena on siis tehdä korkean tason kartoitus, jonka jälkeen valitaan tarkemmin prosesseja tai prosessin osia, joita hallinnoimalla päästään haluttuun ratkaisuun. (Dumas et al. 2013, 16–22) Jos yritys on esimerkiksi havainnut asiakkaiden tekemien reklamaatioiden määrän nousseen merkittävästi, se voi ruveta tarkastelemaan ongelmaa etsimällä, mitä tuotteita nämä reklamaatiot erityisesti koskevat. Tämän jälkeen yritys voi rajata korjaustöiden piiriin ne linjat, jotka valmistavat näitä tiettyjä tuotteita. Nyt koko liiketoimintaa koskeva ongelma on rajattu muutamaan prosessiin ja ongelmanratkaisu helpottuu.

Toisessa vaiheessa valittu prosessi tai tehtävä kuvataan tarkemmin sen nykyisessä muodossa. Jokainen tehtävä prosessissa mallinnetaan usein liiketoiminnan tason (engl. business level) tarkkuudella verrattuna aikaisemman vaiheen korkeatasoisempaan mallinnukseen. Kuvaus yleensä tehdään sekä sanallisesti että käyttäen jotakin visuaalista prosessimallinnuksen kuvaustapaa. Tällä tavalla vältytään epätarkkuuksilta ja tulkintaeroilta liittyen prosessin kulkuun. (Dumas et al. 2013, 16–22; Ruževičius et al. 2012) Tässä vaiheessa esimerkin yritys voi mallintaa ongelmallisten linjojen toimintaa käyttäen esimerkiksi BPMN-syntaksia. Mallien tulisi kuvata mahdollisimman totuudenmukaisesti, miten prosesseissa toimitaan, jotta mahdolliset epäkohdat saataisiin esille.

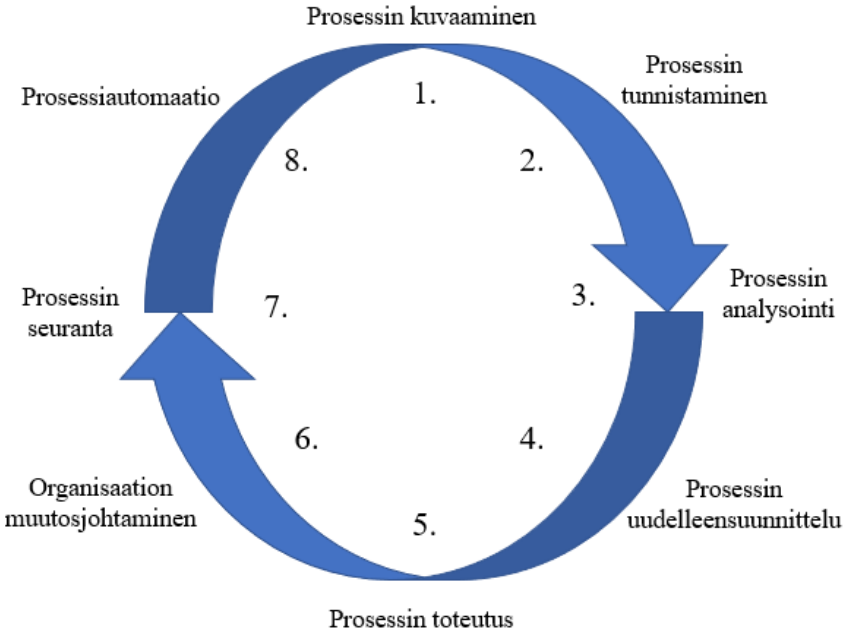
Seuraavaksi toisessa vaiheessa tehty malli analysoidaan ja siinä löydettävät parannuskohteet kirjataan ylös ja priorisoidaan. Jos mahdollista, ongelmille voidaan myös antaa jonkinlainen määrällinen kuvaus, kuten kuinka usein jossain tietyssä tehtävässä tapahtuu keskeytyksiä viikon sisällä. (Dumas et al. 2013, 16–23) Esimerkkiyritys voi tässä kohdassa tarkastella prosessin laadunvalvonnan toteutusta mallinnuksen kautta ja arvioida, onko nykyinen toimintatapa riittävä suhteessa tavoiteltuun laatutasoon.

Analyysin jälkeen ruvetaan suunnittelemaan erilaisia parannuskeinoja prosessille. Tässä kohtaa tulisi tietää, mikä on organisaation haluama taso tälle prosessille ja miten sitä voidaan mitata, jotta varmistutaan, että tavoitteisiin on päästy. Parannusvaihtoehdoista valitaan kaikkein lupaavin ja sen toteutusta suunnitellaan tarkemmin. Tässä kohdassa kannattaa usein

myös tehdä mallinnus ja/tai sanallinen kuvaus prosessin mahdollisesta tilasta, kun parannusmuutokset on toteutettu, sillä sen vertaaminen nykytilan malliin auttaa hahmottamaan paremmin ne kohdat prosessia, joihin tulisi kohdistaa eniten parannuksia tai muutoksia. (Dumas et al. 2013, 16–23; Ruževičius et al. 2012)

Seuraavaksi valittu parannuskeino toteutetaan, jotta päästään aikaisemmassa vaiheessa kuvattuun prosessin ideaalitilaan. Tämän toteutusvaiheen voi jakaa kahteen osaan, jotka ovat organisaation muutosjohtaminen ja prosessiautomaatio. Organisaation muutosjohtamisessa on tarkoitus tiedottaa ja kouluttaa tulevista muutoksista henkilöille, joita tämä muutos tulee koskemaan. Lisäksi tässä vaiheessa tulisi tehdä suunnitelma muutoksen läpiviennistä. Prosessiautomaatio taas viittaa parannussuunnitelmaan liittyvien tietotekniikan osien valmisteluun, kuten uuden ohjelman käyttöönotto tai ohjelmistorobotin ohjelmointi. Nykyään on myös mahdollista rakentaa automaatioratkaisuja suoraan suunnitellun mallinuksen päälle, mutta toteutuksen ja suunnittelun vaiheet tulisi silti pitää erillään prosessinjohtamisen selkeyden takia. (Dumas et al. 2013, 16–23; Ruževičius et al. 2012)

Viimeisenä muttei vähäisempänä vaiheena on prosessin seurantaan liittyvät tehtävät. Varsinkin ratkaisun käyttöönoton alussa ilmenee varmasti ongelmia ja puutteita, jotka on saatava korjattua. Kun ratkaisu toimii kuten sen kuuluukin, olisi hyvä mitata määrällisesti prosessin kulkua verrattuna vanhaan, jotta konkreettiset hyödyt saadaan näkyville. Pitemmällä välillä prosessia ylläpidetään, kunnes tulee aika, jolloin se on päivitettävä uudelleen ja prosessijohtamisen elinkaariprosessi alkaa uudelleen (Kuva 3). Jos tämä viimeinen vaihe jätetään tekemättä, ei prosessijohtamisen elinkaari toteudu. (Dumas et al. 2013, 16–24; Ruževičius et al. 2012) Esimerkkiyrityksen tulisi jatkossakin seurata prosessiensa tuloksia ja toimia aktiivisesti niiden parantamisessa.



Kuva 3. Prosessijohtamisen elinkaari on jaksottainen prosessi

Kuten Kuvasta 3 nähdään, prosessijohtaminen on jaksottainen ja jatkuva prosessi, jossa toimintaa on tarkoitus kehittää jatkuvasti eteenpäin. Siinä korostuu yhteistyö organisaation tietotekniikan ja liiketoiminnan osaajien välillä, sillä kumpaakin tarvitaan, jotta prosessijohtamisesta saataisiin kaikki irti. Prosessijohtamisen elinkaarta voidaan käyttää myös hyödyksi automaatioprojekteissa, sillä prosessiautomaatio on rakennettu osaksi tätä prosessia. Automaatioprojekti tietysti vaatii myös omia lisätietoja ja -valmisteluja, joita ei käydä prosessijohtamisen elinkaaressa. (Dumas et al. 2013, 16–24; Mendling 2022)

2.2 Prosessin suorituskyvyn mittaaminen

Prosesseille voidaan asettaa erilaisia mittareita niiden suorituskyvyn selvittämiseksi. Ne ovat myös tärkeässä roolissa prosessijohtamisen elinkaaressa. Tärkeitä mittareita prosessille kuin prosessille ovat aikaan, kustannuksiin, laatuun ja joustavuuteen liittyvät mittarit. Niiden alle mahtuu myös muita mittareita ja erilaisia mittaustapoja. Suorituskyvyn mittarin tulisi olla yksiselitteinen ja tietystä prosessista kerättäviin lukuihin perustuva sekä jonkinlainen jalostus (engl. aggregation) näistä luvuista, esimerkiksi keskiarvo tai prosenttiosuus. Yrityksen

sisällä tulisi tunnistaa selvät syy-seuraussuhteet käytettyjen mittareiden ja seurattavien prosessien välillä. (Dumas et al. 2013, 59–60; Rantanen & Holtari 1999)

Aikaan liittyvistä mittareista sykliajan lyhentäminen eli suoritusteho on todennäköisesti se mittari, johon pyritään vaikuttamaan kaikkein eniten prosessijohtamisen ja -automaation avulla. Myös sykliajan vaihtelevuutta voidaan haluta vähentää, jotta toiminnasta tulisi johdonmukaisempaa ja jotta toiminta esimerkiksi toimittajien kanssa sujuisi paremmin. Sykliaika voidaan vielä jakaa kahteen osatekijään: palveluaikaan ja odotusaikaan. Palveluajalla tarkoitetaan prosessin työstämiseen kuluvaa todellista aikaa, kun taas odotusajalla tarkoitetaan kaikkea sitä aikaa, joka prosessissa käytetään resurssien, muiden prosessien tai ulkopuolisten syötteiden odottamiseen tai jonottamiseen. Palveluaikaa halutaan siis tehostaa mahdollisimman paljon ja samaan aikaan odotusaika halutaan minimoida. (Amaratunga & Baldry 2002, Dumas et al. 2013, 59–60)

Taloudelliset suorituskyvyn mittarit keskittyvät yleensä kustannusten leikkaamiseen tehostustoimien kautta. Kustannukset voidaan jakaa kahteen pääkategoriaan eli muuttuviin ja kiinteisiin kustannuksiin. Muuttuvat kustannukset ovat suoraan riippuvaisia prosessin kulusta, kuten työntekijöiden tai hankittujen materiaalien määrästä, siten muuttuviin kustannuksiin voidaan vaikuttaa suoraan parantamalla prosessin suorituskykyä. Kiinteät kustannukset taas eivät riipu suoraan prosessin suorituskyvystä, vaan ne mahdollistavat olosuhteet prosessin olemassaololle esimerkiksi rakennusten ja ohjelmistojen lisenssimaksujen kautta. Kiinteisiin kustannuksiin ei siis yleensä voida vaikuttaa, mutta prosessin tehostaminen lähemmäs kapasiteettia tehostaa myös kiinteiden kustannusten suhteellista käyttöä. Yksittäistä prosessia mitattaessa puhutaan usein toimintakustannuksista, jotka sisältävät prosessin tuotokseen liittyvät kustannukset. Taloudellisen suorituskyvyn mittarit ovat usein suoraan sidottuja tilinpäätöksen lukuihin, joten ne ovat myös ulkoisille toimijoille näkyviä, jonka takia ne halutaan esittää mahdollisimman hyvinä. Yrityksillä on kuitenkin sisäisesti paljon muutakin dataa ja tietoa käytettävissään, joten niiden ei pitäisi jumittua katsomaan ainoastaan taloudellisia mittareita. (Dumas et al. 2013, 60; Rantanen & Holtari 1999)

Laadullinen suorituskyky voidaan mitata joko asiakkaan tai organisaation sisäisestä näkökulmasta, jolloin puhutaan ulkoisesta ja sisäisestä laadusta. Ulkoinen laatu kuvaa sitä, kuinka hyvin ollaan tuotteen tai palvelun osalta onnistuttu täyttämään asiakkaan toiveet ja tekniset tai toiminnalliset vaatimukset. Sisäinen laatu kuvaa taas prosessin onnistumista siihen osallistuvan henkilön näkökulmasta. Tekijät kuten prosessin suorittamisen yhdenmukaisuus, suoraviivaisuus ja hallittavuus halutaan tyypillisesti mahdollisimman korkeiksi. Laatua mitataan usein aikaan ja kustannuksiin liittyvillä mittareilla, jolloin syntyy jonkin verran päällekkäisyyksiä. (Dumas et al. 2013, 60–61; Kinney & Rayborn, 508)

Prosessin joustavuus kuvaa prosessin suorituskykyä normaalitilanteen ulkopuolella esimerkiksi erityistilanteissa, kuten ruuhka-aikoihin. Yleisesti joustavuuden voidaan ajatella olevan prosessin kyky sopeutua erilaisiin kapasiteetin rasituksiin. Joustavuutta voidaan mitata liiketoiminnan eri osa-alueilta kuten resurssien käytön joustavuudessa, prosessin kyvyssä sopeutua äkillisiin muutoksiin ja hallinnon tekemiin rakenteellisiin muutoksiin prosessin ympärillä tai sisällä. Vaikka joustavuus on erittäin tärkeä prosessin suorituskyvyn mittari, sen mittaaminen on haastavaa ja usein organisaatioiden on kehitettävä omia mittareita sen arvioimiseksi. (Dumas et al. 2013, 61) Taulukkoon 2 on koottu yhteen aiemmin käsitellyt suorituskyvyn mittarit.

Taulukko 2. Yhteenveto prosessin suorituskyvyn perusmittareista sekä niiden oleellisimmista osa-alueista

Ajallinen suorituskyky	**Taloudellinen suorituskyky**	**Laadullinen suorituskyky**	**Prosessin joustavuus**
Sykliajan yhdenmukaisuus	Muuttuvat kustannukset	Ulkoinen laatu	Suorituskyky poikkeustilassa
Palveluaika	Kiinteät kustannukset	Sisäinen laatu	Resurssien joustavuus
Odotusaika	Toimintakustannukset	Laatu suhteessa aikaan ja kustannuksiin	Rakenteellinen joustavuus

Tässä osiossa käsitellyt mittarit ovat hyvin yleisiä ja ne kiinnostavat prosessissa kuin prosessissa. Tämä diplomityö on erityisen kiinnostunut ajallisen, taloudellisen sekä laadullisen

suorituskyvyn parantamisesta toimihenkilötyössä. Toimihenkilöiden tekemä työ on kuitenkin jonkin verran haastavampaa mitata, jota käsitellään tarkemmin seuraavassa osiossa.

2.3 Toimihenkilötyön määritelmä ja mittaus

Toimihenkilölle kuuluva työ ja vastuu ovat asioita, jotka vaihtelevat vahvasti aikakaudesta, näkökulmasta ja alasta riippuen. Esimerkiksi juridisesta näkökulmasta toimihenkilöihin kuuluvat henkilöt, joiden työsopimuksessa sovelletaan toimihenkilöiden tai ylempien toimihenkilöiden työehtosopimusta. Käytännössä nykyään toimihenkilötyö mielletään ainakin suomalaisissa yrityksissä erilaisiksi asiantuntija-, tutkimus- ja johtotasontöiksi. (YTN 2023; Tilastokeskus 2023) Tässä diplomityössä toimihenkilötyöllä tarkoitetaan pääasiassa teknologiateollisuuden yrityksen keskijohdon sekä näiden alaisuudessa toimivien asiantuntijoiden työtä. Töihin kuuluu muun muassa strategisen hankinnan, operatiivisen oston sekä laatu- ja suunnittelupäällikön vastuualueille kuuluvia töitä.

Toimihenkilöiden tuottavuuden mittaaminen on aina ollut suhteellisen hankalaa verrattuna esimerkiksi tavallisten työntekijöiden suoritusten mittaamiseen. Perinteinen tuotantotyö voidaan yksinkertaisimmillaan mitata vertaamalla prosessin käyttämien resurssien arvoa siitä ulostulevien tuotteiden arvoon. Työn tehokkuutta voidaan myös mitata ajallisesti, esimerkiksi keskimääräisenä projektin pituutena. Toimihenkilötyö on luonteeltaan asiantuntija- ja tiedontuotantotyötä ja se tehdään usein osana ryhmää, jolloin arvon määrittely muuttuu haastavammaksi. Tuotantotyötä on pystytty tehostamaan huomattavan paljon hyödyntäen erilaisia teknologioita ja automatiikkaa, kun taas toimihenkilötyö on pysynyt tehokkuuden kehitykseltään hyvin tasaisena. Tämä on erityisen huomattavaa, sillä työmarkkinakehitys, ainakin Suomen kaltaisissa varakkaissa länsimaissa, on pitkään ollut siirtymässä pois perinteisestä tuotantotyöstä kohti palvelu- ja asiantuntijatöitä, jotka kuuluvat toimihenkilötyön piiriin. (Ahmed et al. 1991, 1–3)

Ensimmäisenä organisaation on ymmärrettävä, ettei toimihenkilötyötä pystytä mittaamaan yhtä tarkasti kuin tuotantotyötä eikä tuloksia tule esittää vertailukelpoisina. Toisekseen toimihenkilöt on saatava osallistumaan oman työnsä mittaukseen ja seurantaan. Kuitenkin työn

suorituskyvyn mittaamiseen liittyvät perusideat, joita käsiteltiin edeltävässä alaluvussa, pätevät suurilta osin myös toimihenkilötyöhön. Kun lähdetään suunnittelemaan toimihenkilötyön mittaamista, on hyvä ensin ottaa huomioon, miltä tasolta suorituskykyä halutaan tarkastella. Jos tuottavuutta tarkastellaan organisaation tasolta, tulisi organisaation kaikkien toimihenkilöiden kuluja verrata esimerkiksi tuotteiden ja palveluiden tuomiin tuottoihin. Tällä tavalla saadaan hyvä yleiskuva esimerkiksi johdon käyttöön, mutta mittari ei ole kovin tarkka. Asteen alemmalla tasolla voidaan tarkastella osasto- tai funktiokohtaisia suorituskykyjä. Toimihenkilöt voidaan jakaa ryhmiin heidän tehtäviensä samankaltaisuuksien mukaan, mutta jokaisen ryhmän osuus yhteisestä tuotoksesta ei ole aina selvä. Funktioitten osuus jostain tuotoksesta on usein selvempi, mutta funktion sisäisten toimihenkilöiden työ ei usein ole täysin homogeenistä ja siten ei kovin vertailukelpoista. Alimpana tasona on yksittäisten toimihenkilöiden suorituskyvyn mittaaminen. Tällä mittaustavalla tulevat esille toimihenkilöiden erilaiset vastuut ja työalueet, mutta mittarin tulisi myös olla dynaaminen, sillä toimihenkilön työnkuva on harvoin täysin pysyvä. Mittarin toteutus tulisi aloittaa kyseisten toimihenkilöiden tiedottamisella sekä tarkalla tiedonkeruulla heidän työtehtävistään. Oleellinen osa mittariston toteutusta on myös tiedon julkaisu sellaisessa muodossa, jossa henkilöt pystyvät ymmärtämään ja vaikuttamaan niihin. Tavoitteet on myös asetettava sellaisille tasoille, että ne ovat realistisesti tavoitettavissa. (Ahmed et al. 1991, 2–9)

2.4 Liiketoimintaprosessien mallintaminen

Mallinnuksella pyritään kuvaamaan monimutkaisia ilmiöitä tai prosesseja todellisesta maailmasta yksinkertaisemmalla mutta silti todenmukaisella tavalla. Prosessimallinnusta tarvitaan muun muassa, kun halutaan lisätä ymmärrystä jostain ilmiöstä, kun halutaan löytää kehittämiskohteita jostain kohdealueesta, kun halutaan yhdenmukaistaa jotakin suurta kokonaisuutta, kun halutaan tuoda automaatiota johonkin työvaiheeseen tai kun halutaan seurata jotakin toimintaa. (Luukkonen et al. 2012) Liiketoimintaprosessien mallinnus (engl. Business Process Modeling) on kokoelma työkaluja, menetelmiä ja tekniikoita systemaattiseen toiminnan kuvaukseen ja tehostamiseen. Prosessimallinnus on myös oleellinen osa prosessijohtamisen elinkaarta, sillä se on tehokas keino esittää selkeä ja todenmukainen kuva yrityksen tilanteesta (Dumas et al. 2013, 16–23). Nykyisten tehokkuutta vaativien markkinaolosuhteiden johdosta sekä työnkulkuohjelmistojen paremman saatavuuden ansiosta,

liiketoimintaprosessien mallinnusta pystytään hyödyntämään yhä useammanlaisissa yrityksissä (Bradford & Dumas 2007).

Prosessimallinnuksessa on tunnettava mallinnettavan prosessin tyyppi hyvin ennen kuin voidaan edes päättää mallintamismetodi. Yrityksistä voidaan usein löytää sekä staattisia että dynaamisia prosesseja joskus myös sekaisin. Staattinen prosessi voi esimerkiksi olla hyvin standardoitu tuotteen rakentamis- ja kokoamisprosessi, kun taas asiantuntija- ja suunnittelutyö nähdään enemmän dynaamisempina prosesseina. Yleisesti kuitenkin prosessi määritellään joukkona toisiinsa liittyviä tapahtumia tai vaiheita, joilla pyritään yhteiseen päämäärään, joka eroaa prosessin lähtötilasta. Resurssit, olivatpa ne sitten taloudellisia, aineellisia tai aineettomia, ovat myös oleellinen osa prosessien määritelmää. (Luukkonen et al. 2012; Dumas et al. 2013, 17–20) Prosessin tarvitsemat resurssit usein määrittävät, mikä on paras tapa mallintaa ja mitata niitä. Esimerkiksi dataintensiivinen prosessi saattaa tarvita erityisesti datan ja tiedon liikkeiden kuvaamiseen tarkoitettuja mallinnusjärjestelmiä. Mallinnusprosessissa voidaan nähdä yksinkertaisimmillaan olevan viisi vaihetta:

1. Tiedon keruu
2. Luonnosvaihe
3. Mallinnus
4. Validointi
5. Jakelu

Mallintamista aloitettaessa on huomioitava, mitkä kokonaisuudet kuuluvat prosessin piiriin ja miten nämä kokonaisuudet vaikuttavat suhteessa toisiinsa sekä vähintään yhtä tärkeästi, mitkä asiat rajautuvat pois kokonaan prosessin mallintamisesta. On myös tärkeää tietää etukäteen malliin tarvittava tarkkuus ja mahdolliset käyttö- ja yhteensopivuustarkoitukset, jotta mallia voidaan hyödyntää yrityksen sisäisillä työkaluilla tai päivittää myöhemmin. Jos organisaation sisällä on käytetty tiettyä mallinuksen tapaa, tyyliä tai merkintätapaa, on syytä huomioida myös tämä ennen uuden mallin aloittamista. Konteksti mallin ympärillä on myös tärkeää. Joskus pienestä prosessin osasta tarvitaan tarkemmin määritelty malli, mutta sitä ympäröivät prosessin osat on jotenkin saatava myös kuvattua, jotta alkuperäisen prosessin osan rooli ja tärkeys säilyvät. (Luukkonen et al. 2012)

2.5 Prosessimallinnuksen erilaiset näkökulmat

Tärkeintä mallinnuksessa on kuitenkin se, että tuloksena syntynyt malli on sitä tarvitsevien henkilöiden saatavilla ja ymmärrettävissä. Tästä syystä on oleellista tietää, kenen käyttöön malli on tulossa ja mitä tämä henkilö tai ryhmä tavoittelee mallinnuksen kanssa. Prosessimallinnuksen yleisimmät näkökulmat, niiden tavoitteet, niiden tuotokset ja niiden tarkoitukset on kuvattu Taulukossa 3.

Taulukko 3. Prosessimallinnuksen näkökulmat ja ominaisuudet (per. Luukkonen et al. 2012)

Näkökulma	Tavoite	Tuotettu malli	Kuvauksen tarkoitus
Johto	• Organisointi • Tehostaminen • Yhdenmukaistaminen • Tulosten parannus • Laadunvalvonta • Seuranta	• Ylätason prosessit • Keskeiset liiketoimintaprosessit • Arvoverkot • Arvon kertyminen liiketoiminnassa	• Strategisen ja taktisen tason suunnittelu ja päätöksenteko • Operatiivinen johtaminen • Toiminnan kehittäminen ja tehostaminen
Työntekijä	• Toimintaohje • Perehdytys • Parannuskohteiden tunnistaminen • Työn tietotarpeiden tunnistaminen	• Prosessien kuvaukset • Työnkulku prosessissa • Työn tehtävät ja niiden järjestys	• Tietojärjestelmän ja työntekijöiden välisen vuorovaikutuksen selvitys, kuvaus ja kehittäminen • Työntekijöiden tai ryhmien välisen yhteistyön ja kommunikoinnin parantaminen
Kehittäjä	• Automatisointi • Prosessin tukeminen ohjelmistoilla • Kohdealueen ymmärtäminen • Simulointi	• Työnkulku prosessissa • Prosessien toiminnot • Työnkulkujen suhde toisiinsa • Rajapinnat • Esi- ja jälkiehdot • Heräte (event/trigger) • Syötteet (input) • Tulokset (output)	• Huomio eri näkökulmien tarpeista • Mallista ilmenevät mahdollisuudet ja rajoitteet • Tiedonkulku • Arkkitehtuurirajoitteiden tunnistaminen
Asiakas	• Toimintaohje • Perehdytys	• Työnkulku prosessissa • Esittely toiminnasta ulkoisille toimijoille	• Riippuu asiakkaan tyypistä ja näkökulmasta

Tässä diplomityössä hyödynnetään kehittäjän näkökulmaa mallinnuksessa, sillä työn tarkoituksena on automatisointi, prosessin tukeminen ohjelmistoilla sekä ympäröivän prosessin parempi ymmärtäminen. Myös joitain johdon tavoitteita kuten laadunvalvontaa ja tehostamista tavoitellaan tässä työssä. Kehittäjän näkökulmasta tehdyn mallin tulisi sisältää työnkulun logiikka sekä toiminnot, erilaiset rajapinnat esimerkiksi ohjelmistojen välillä, ehdot tehtävien aloittamiselle ja täyttymiselle, heräte prosessin aloitukselle ja tarvittavat syötteet sekä odotettavat tulokset. Valittu prosessi voidaan aluksi kuvata mustana laatikkona, joka lähtee herätteestä liikkeelle ottaen vastaan tarvittavat resurssit, ohjauksen ja syötteet tuottaen niillä jonkin tuloksen. Tämä musta laatikko voidaan myöhemmin avata tarkemmassa kuvauksessa, jossa saadaan näkyville sen yksittäiset toiminnot ja tehtävät.

2.6 Mallin kuvaustavan ja notaation valinta

Se millä tavalla prosessi kuvataan ja mitä notaatiotyyliä siinä käytetään vaikuttaa vahvasti lopputulokseen ja sen näyttämään tietoon. Oikean kuvaustavan valinta on tärkeää, jotta lopullisesta mallista ei jää pois mitään oleellista tietoa tai toiminnallisuutta. Mallin näkökulma, tarkkuustaso, mahdollinen välinetuki ja yleinen tavoite yrityksessä on tiedettävä ennen kuin kuvaustapa tai notaatio valitaan. Joitain esimerkkejä tällaisista merkintätavoista ovat BPMN-syntaksi, UML-aktiviteettikaaviot, Little-JIL, YAWL sekä yleisemmät ja vähemmän muodolliset prosessilistat ja -taulukot. (Luukkonen et al. 2012; Dumas et al. 2013 17–19)

Tässä diplomityössä mallinuksen syynä on halu selvittää tietyn prosessin kulkua, jotta sitä voidaan analysoida ja tehostaa automaatiolla. Näin ollen mallin on oltava suhteellisen tarkka ainakin automatisoitavan prosessin osan osalta ja sen kuuluu perustua tosi maailmaan eikä esimerkiksi tavoitetilaan. Kaikkein korkeimpia tasoja ei siis ole tarpeen kuvata kovin tarkasti, mutta konteksti prosessille on säilytettävä. Kuitenkin, jos malliin perustuvaa prosessia joskus muutetaan, on mallia muutettava tämän mukaisesti. Mallista tulisi myös näkyä mahdolliset vika- ja poikkeustilanteet. Tässä mallissa hyödynnetään kehittäjän näkökulmaa, sillä tarkoituksena on testata automaation vaikutuksia tarjous-hankintaprosessin suorituskykyyn. Mallissa halutaan kuvata prosessin työnkulkua, jotta toimihenkilötyön automaation

potentiaalia voitaisiin arvioida ja jotta oikea sijoituspaikka löytyisi sille. Myös rajapintojen kuvaaminen eri prosessien ja toimijoiden välillä korostuvat tässä tarkoituksessa. Näihin tekijöihin sekä diplomityön rajoituksiin perustuen on valittava sopivin kuvaustapa tai notaatio.

UML (Unified Modeling Language) on operationaalinen kuvaustapa, joka soveltuu erityisen hyvin tarkasti määriteltyjen prosessien tai yksittäisten tehtävien kuvaukseen. Se on kielenä varsin joustava varsinkin yrityksen alempia tasoja ja toimintoja mallinnettaessa. UML-kieli kehitettiin yhtenäistämään ohjelmistoarkkitehtuuria kuvaavia notaatioita, mutta sitä voidaan myös käyttää muunlaisissa konteksteissa. UML-kieli on suuri ja monimutkainen, eikä sitä kannata hyödyntää esimerkiksi suurten yritysprosessien mallintamisessa ainakaan yksin. Näistä syistä UML-kieltä ei tulla käyttämään tässä diplomityössä, mutta sen käyttöä voidaan harkita projektin diplomityön jälkeisessä osassa. (Luukkonen et al. 2012; Object Management Group 2017)

Little-JIL on toinen kuvauskieli, joka sopii erityisesti tarkempaan operationaaliseen kuvaukseen. Tämä kuvaustapa kehitettiin alun perin määrittämään ohjelmistojen kehitystä ja ylläpitoa. Little-JIL-kielessä on määriteltävä saatavilla olevat artefaktit, resurssit, toimijat sekä työvaiheet ja niiden järjestys sekä saatavuus. Little-JIL-kielessä voidaan tehdä sisäkkäisiä kuvauksia tarkempien kuvausten aikaansaamiseksi. Oleellisia vahvuuksia Little-JIL-kielessä ovat sen kyky määrittää ajoituksia, samanaikaisia prosesseja sekä kyky määritellä tarkasti toimintojen rajapintojen tieto- ja dataliikenne. Nämä tekijät avustaisivat automaation suunnittelussa ja toteutuksessa, mutta prosessien kulku ei välttämättä aukea tavalliselle business-käyttäjälle, mikä vaikeuttaisi validointivaihetta. Lisäksi yrityksen ohjelmistovalikoimasta puuttuu tätä kieltä tukeva piirto-ohjelma, joka työn rajoitusten mukaisesti rajaa tämän kielen pois. (Luukkonen et al. 2012; Zhu et al. 2007)

YAWL (Yet Another Workflow Language) on avoimeen lähdekoodiin perustuva kuvauskieli, joka on erikoistunut erilaisten työnkulkumallien kuvaamiseen. Sillä on jonkin verran päällekkäisyyksiä muiden työnkulkua kuvaavien notaatioiden kuten BPMN- ja UML-kielien kanssa. YAWL kuitenkin eroaa niistä sen kyvyllä määrittää tarvittavia resursseja ja aikaa tiettyihin tehtäviin. YAWL-mallit ovat myös enemmän verkostomaisia kuin lineaarisia

työnkulkuja, joita näkee esimerkiksi BPMN-malleissa. YAWL-ohjelmistoilla on myös kyky suorittaa tehtäviä, mikä tekisi siitä ihanteellisen mallinnusohjelman automaatioprojekteille. (Luukkonen et al. 2012; Bradford & Dumas 2007) Valitettavasti yrityksen ohjelmistopakettiin ei kuulu yhtään YAWL-kieltä tukevaa ohjelmaa, joka poistaa tämän kielen harkinnasta tässä diplomityössä. On kuitenkin mahdollista, että jos tämän diplomityön aloittamaa projektia jatkokehitetään, YAWL saattaa olla harkitsemisen arvoinen työkalu.

BPMN (Business Process Modeling Notation) on prosessien ja tehtävien kuvaamistapa, joka keskittyy erityisesti työn etenemisen logiikan selkeään kuvaukseen. BPMN-syntaksin etuihin kuuluu sen yleisyys erilaisissa piirto-ohjelmissa sekä sen standardisoitu esitystapa, sillä samoin kuin UML-kieli, BPMN-notaatio on myös OMG:n (Object Management Groupin) hallinnoima ja päivittämä. BPMN 2.0.2 on uusin versio tästä syntaksista ja se sisältää muun muassa mahdollisuuden ladata metaversio visuaalisesta mallista sekä toiminnallisen semantiikan mallin osille, mikä luo paremmat mahdollisuudet automaation sisällyttämiselle malliin verrattuna aikaisempiin versioihin. Tosin vaikka tämä potentiaali on olemassa, vain harvat ohjelmistotuottajat tukevat sitä. BPMN-syntaksilla tehdyt mallit ovat suhteellisen helppolukuisia myös henkilöille, joilla ei ole aikaisempaa kokemusta sen notaatiotyylistä ja samaan aikaan sillä pystytään mallintamaan myöskin monimutkaisia kokonaisuuksia ja kevyesti myös tiedon liikkumista. (Luukkonen et al. 2012; Silver 2011; Object Management Group 2014) Muun muassa nämä ominaisuudet tekevät BPMN-notaatiosta hyödyllisen tässä diplomityössä, jossa mallin toiminnan selittäminen prosessin asiantuntijoille on oleellisessa osassa. Yrityksellä on myös valmiina käytössä ohjelmistoja, jolla BPMN-mallinnusta voidaan tehdä.

3 Prosessiautomaatio

Viime vuosina on nähty huomattava määrä mullistavia teknologioita, joilla on ollut potentiaalia muuttaa monien toimialojen toimintatapoja. Esimerkkejä tällaisista teknologioista ovat tekoäly (engl. Artificial Intelligence), esineiden internet (engl. Internet of Things) ja lohkoketju (engl. Blockchain). Nämä teknologiat ovat osa niin kutsuttua neljättä teollista vallankumousta (engl. Industry 4.0), johon liitetyt ratkaisut tyypillisesti yhdistävät fyysisiä ja virtuaalisia ympäristöjä sekä hyödyntävät reaaliaikaista dataa. (Singh et al. 2019) Prosessijohtaminen tässä ympäristössä on muuttunut haastavammaksi, kun yhä useampi organisaatio haluaa hyödyntää näitä monimutkaisia teknologioita ja samaan aikaan optimoida jo käytössä olevaa legacy-teknologiaansa. Prosessiautomaatio ja sen työkalut voidaan nähdä myös osana tätä teknologia-aaltoa. (Ahmad & van Looy 2020)

3.1 Liiketoimintaprosessien automaatio

Liiketoimintaprosessia voidaan kuvata sarjana toisiinsa liittyviä tehtäviä, jotka yhdessä luovat jonkin yhtenäisen tuloksen. Osa tai jopa kaikki näistä tehtävistä voivat kuitenkin olla sellaisia, joita ihmisten ei tarvitsisi tehdä. Esimerkiksi laskun manuaalinen lähettäminen sähköpostilla asiakkaalle tai tiedon siirtäminen yhdestä rajapinnasta toiseen käsin ovat tehtäviä, jotka vievät aikaa henkilön varsinaisesta työstä. Toimisto- ja toimihenkilötyössä tällaiset turhat toimitukset korostuvat erityisesti ja yhteen laskettuna ne vievät huomattavasti aikaa varsinaisesta ajattelu- ja tiedontuottotyöstä, jota toimihenkilöiden olisi tarkoitus tehdä. Muun muassa informaatioteknologian kehitys ja digitalisaatio ovat olleet suurina vaikuttajina liiketoimintaprosessien automaation mahdollistamisessa. Myös yritysten halukkuus kerätä dataa ja tietoa sisäisistä toimistaan ja luoda malleja liiketoiminnastaan ovat osatekijöitä automaation käyttöönoton nopeutumisessa ja yleistymisessä. (Hyun et al. 2021) Seuraavissa luvuissa käydään läpi automaatioon liittyvien käsitteiden määritelmiä, automaatioon liittyviä vaatimuksia, työkaluja ja toteutuksen toimenpiteitä sekä automaation tuloksia.

Liiketoimintaprosessien automaatio voidaan määritellä yksinkertaisesti siten, että ihmisen aikaisemmin tekemä tehtävä korvataan ohjelmiston tai ohjelmistorobotin suorituksella. Liiketoimintaprosessien automaatio voi vaihdella hyvin yksinkertaisista ja lyhyistä ohjelmista koko prosessin kattaviin kokonaisuuksiin, joissa ihmisen vuorovaikutus on minimaalista. Automaatioprojektin laajuus on vahvasti sidoksissa kohdeprosessin rakenteeseen ja tarpeisiin. (Hyun et al. 2021) Tehtävät, jotka soveltuvat erityisen hyvin automaatiolle ovat luonteeltaan toistuvia, arkipäiväisiä ja niillä on tarkasti määritellyt säännöt (Santos et al. 2020). Tehtävien on oltava standardisoituja ja tarkasti rajattuja, jotta tulosten ja läpiajojen vaihtelevuutta saadaan rajoitettua. Jos prosessin aikana täytyy tehdä päätöksiä, niiden tulisi olla yksiselitteisiä ja mielellään binäärisiä. Ihmisharkinnan käyttö automaatioprosessissa on mahdollista, sillä ohjelmistorobotti voi kysyä esimerkiksi ponnahdusikkunan kautta ihmisen päätöstä jostakin. Tämä kuitenkin hidastaa prosessin kulkua ja on käänteinen koko prosessiautomaation tarkoitukselle ja lisäksi se vaatii robotin ja ihmistyön synkronointia, joka ei aina ole mahdollista tai kannattavaa. Tästä syystä tehtävät, jotka vaativat paljon analyysiä ja/tai ihmisharkintaa eivät sovellu hyvin tai lainkaan automaatiolle. Automatisoidun tehtävän tulisi myös olla usein tai erittäin usein toistuvaa, jotta siitä saadaan mahdollisimman suuri hyöty ja nopea takaisinmaksuaika. Myös suurten jäsenneltyjen data- tai tietomäärien liikkuminen tehtävän aikana tekee tehtävästä paremmin soveltuvan automaatiolle varsinkin silloin kun kyseessä on rutiininomainen datan syöttö tai datan siirtäminen eri ohjelmistorajapintojen välillä. Korkealla organisaatiotasolla tulisi myös tarkastella, kuinka usein tämä tehtävä on muuttunut ja mikä sen arvo on yritykselle. Automaatiolle sopivimmat tehtävät ovat muuttumattomia tai erittäin harvoin muuttuvia ja organisaation toiminnan kannalta strategisia. (Wellman et al. 2020; Fung 2014; Jimenez-Ramirez et al. 2019; Wanner et al. 2019; Yatskiv et al. 2019)

3.2 Ohjelmistorobotiikan kuvaus

Ohjelmistorobotiikka (engl. Robotic Process Automation, RPA) on yksi yleisimmistä liiketoimintaprosessien automaatiossa käytetyistä työkaluista. Siinä jokin tehtävä tai prosessi on korvattu automaattisesti toimivalla ohjelmalla, joka toimii tietokoneen käyttöliittymän kautta tavallaan kuten ihminenkin. (van der Aalst et al. 2018; Santos et al. 2020) Tietokoneen ruudulla ohjelmistorobotin toiminta saattaakin näyttää ihmismäiseltä, kun se liikuttaa hiirtä, kirjoittaa tekstiä tai vaihtaa ohjelmiston ikkunaa. Erona ovat kuitenkin robotin nopeus,

tarkkuus ja virheettömyys, kunhan sen toimintaympäristö pysyy vakiona. Syitä ohjelmistorobotiikan yleisyydelle verrattuna muihin automaatiotyökaluihin ovat muun muassa sen joustavuus ja suhteellisen nopea takaisinmaksuaika. Ohjelmistorobotit myös ohittavat tarpeen ohjelmointirajapinnoille (engl. Application Programming Interface, API), mikä tekee niistä huomattavasti helpomman ja joskus ainoan tavan toteuttaa automaatiota organisaatioille, joilla on paljon erilaisia ohjelmistoja käytössä. Myös sopimukset ohjelmistojen tarjoajien kanssa voivat rajoittaa ohjelmistorajapintojen käyttöä. (Kirchmer & Franz 2019; Santos et al. 2020)

Tutkimuksissa on osoitettu ohjelmistorobotiikan toimivan parhaiten, kun sen käytöönotto on yhdistetty systemaattiseen prosessijohtamiseen ja prosessimallinnukseen (Kirchmer & Franz 2019; Kroll et al. 2016; Signavio Whitepaper 2019; Tornbohm 2017). Prosessimallinnukseen sisältyy ohjelmistorobotiikassa tarvittavat tiedot, kuten tehtävien järjestys ja logiikka sekä prosessissa tarvittavat dataobjektit ja ohjelmistot. Myös poikkeusten käsittely voi helpottua integroidulla prosessimallilla, kun mallissa voidaan esimerkiksi värikoodilla näyttää, missä kohdassa ohjelma on pysähtynyt. Prosessijohtamisen ja -mallinnuksen liittäminen ohjelmistorobotiikkaan on myös hyödyllistä pidemmällä aikavälillä, kun prosessiautomaatiota hallinnoidaan koko yrityksen konteksti mielessä, jolloin jatkokehitys on suoraviivaisempaa ja yhdenmukaisempaa. Tiedon keruu, poikkeuksien käsitteleminen ja kokonaisvaltainen automaation johtaminen ovat perinteisesti olleet suurimpia haasteita niin ohjelmistorobotiikassa kun muissakin automaatioprosesseissa. (König et al. 2020)

3.3 Automatisoitavan prosessin valintakriteerien asetus

Tässä osiossa käydään läpi tarkemmin, mitä kannattaa huomioida automatisoitavaa prosessia valittaessa. Aikaisemmin huomioitiin automatisaation rajoitteisiin sopivimpia valintakriteereitä, kun taas tässä harkitaan myös liiketoimintalähtöisempiä kriteereitä. Valintakriteerit yhdistetään taulukkoon osion lopussa. Ensinnäkin tulisi huomioida prosessin strateginen tai muu tärkeys liiketoiminnan kannalta. Prosessiin, joka sisältää paljon riskejä tai epävarmuutta sen pysyvyydestä osana organisaation toimintaa, ei ainakaan kannata kiinnittää automaatioprojektia. Sen sijaan prosessit, jotka ovat strategisesti tärkeitä ja/tai sidoksissa organisaation pitkänaikavälin tavoitteisiin ovat hyviä ehdokkaita tehostamiselle. Tällaiset prosessit voi

tunnistaa esimerkiksi niiden tuomasta uniikista kilpailuedusta sekä niiden potentiaalisesta tai toteutuneesta korkeasta tuloksellisuudesta. Strategisen tärkeyden lisäksi tulisi tarkastella prosessin tilaa, eli toimiiko prosessi jo niin hyvin kuin se voisi toimia. Harkittavat prosessit kannattaa järjestää niiden arvioidun automaatiosta saadun hyödyn määrän mukaan. Lopuksi tulisi myös karkeasti arvioida prosessin parannuksen toteutettavuutta. Jos prosessiin liittyy esimerkiksi epävarmuuksia sen vastuusta tai jos prosessin omistaja ei syystä tai toisesta halua osallistua muutostyöhön, projektiaikaa ei kannata tuhlata kyseiseen prosessiin. On myös arvioitava, riittävätkö organisaatiossa saatavilla olevat resurssit, kuten osaaminen ja työkalut, realistisesti projektin läpivientiin. Tämän kaltaiset kriteerit voidaan saatavilla olevan tiedonmäärän salliessa arvioida määrällisesti mutta myös laadullisesti esimerkiksi johdon ja esimiesten vaikutelmien ja kokemusten perusteella. (Dumas et al. 2013, 56–57) Taulukkoon 4 on yhdistetty nämä liiketoimintakeskeiset sekä aikaisemmin käsitellyt automaatioon liittyvät kriteerit ja rajoitukset.

Taulukko 4. Liiketoimintaan sekä automaatioon liittyvät valintakriteerit ja tekijät

Kriteeri	**Vaikuttaja**	**Osa-alue**
Strateginen merkityksellisyys	Organisaatio	Liiketoiminta
Prosessin suorituskyky	Prosessin kulku	Liiketoiminta
Projektin toteutettavuus	Organisaatio	Liiketoiminta
Prosessin toistuvuus	Prosessin kulku	Automaatio
Prosessin arkipäiväisyys	Prosessin kulku	Automaatio
Prosessin säännönmukaisuus (standardisointi)	Organisaatio	Automaatio
Prosessin tarkka rajattavuus	Prosessin kulku	Automaatio
Yksiselitteiset päätökset (ei inhimillistä harkintaa)	Prosessin kulku	Automaatio
Prosessin data/tieto on jäsenneltyä	Prosessin data/tieto	Automaatio
Prosessiin liittyy datan/tiedon siirtämistä	Prosessin data/tieto	Automaatio
Prosessissa käytetään useita ohjelmia	Prosessin kulku	Automaatio
Prosessin muuttumattomuus	Organisaatio	Automaatio

Taulukosta 4 voidaan nähdä, että valintakriteereitä voidaan löytää useita ja että suurin osa niistä liittyy automaatioon. Myös kriteerien taustalla vaikuttavat tekijät on kerätty

taulukkoon. Prosessin kulku, eli siihen liittyvät tehtävät, päätökset ja logiikka nähtiin yleisimpänä tekijänä kriteerin täyttymiselle esimerkiksi prosessin toistuvuuden, rajattavuuden ja sen käyttämien ohjelmistojen suhteen. Prosessien kulkuun voidaan joissain tilanteissa vaikuttaa, esimerkiksi prosessin suorituskykyä voidaan parantaa tai prosessin käyttämiä ohjelmistoja voidaan karsia. Organisaatiosta riippuvat kriteerit, kuten prosessin strateginen merkityksellisyys tai standardisointi, ovat kriteereitä, joihin ei voida välttämättä vaikuttaa ollenkaan tai niihin vaikuttaminen on hidasta ja haastavaa. Prosessin käyttämään dataan tai tietoon voidaan joissain tilanteissa vaikuttaa ainakin sen jäsentelyn kannalta. Tähän kriteeristöön perustuen voidaan valita oikeanlainen prosessi automaatioprojektia varten.

3.4 Automaatioprojektin toteutus

Automaatioprojektien toteutuksessa automatisoitavan prosessin osan valinta on yksi kriittisimmistä tekijöistä. Sen lisäksi, että väärän prosessin automatisointi vie kehitysaikaa, se myös huonontaa kokonaisprosessin tehokkuutta ja saattaa myös lisätä virheajojen määrää. (Leopold et al. 2018) Automaation toteutuksessa on myös muistettava erot erilaisten automaatiometodien välillä. Perinteisesti prosessiautomaatio on toteutettu ohjelmointirajapintojen avulla, joka saattaa muuttua hyvinkin monimutkaiseksi, kun useamman tuottajan ohjelmistoja on sovitettava yhteen. Usein myös tämän tyyppinen prosessiautomaatio yhdistettynä prosessijohtamiseen vaatii muutoksia liiketoiminnan logiikkaan. Ohjelmistorobotit sen sijaan hyödyntävät tietokoneen käyttöliittymää suorituksessaan, jolloin ohjelmointirajapintoja tai ohjelmiston logiikkaa ei tarvitse muuttaa. Robotit voi myös usein integroida jo olemassa olevan prosessin päälle ilman suuria muutoksia liiketoiminnan kulkuun. Raskaammat ’back-end-automaatioprojektit’ ovat usein IT:n omistuksessa ja vaativat aina osaamista ohjelmoinnissa sekä datan käsittelyssä. Kevyet ’front-end-ohjelmistorobotit’ voivat taas olla niitä käyttävien funktioiden omistuksessa eikä niissä välttämättä tarvita suoraa ohjelmointikokemusta. Erot näiden kahden automaatiomenetelmän välillä on tiivistetty Taulukkoon 5. (Lacity et al. 2016; Santos et al. 2020)

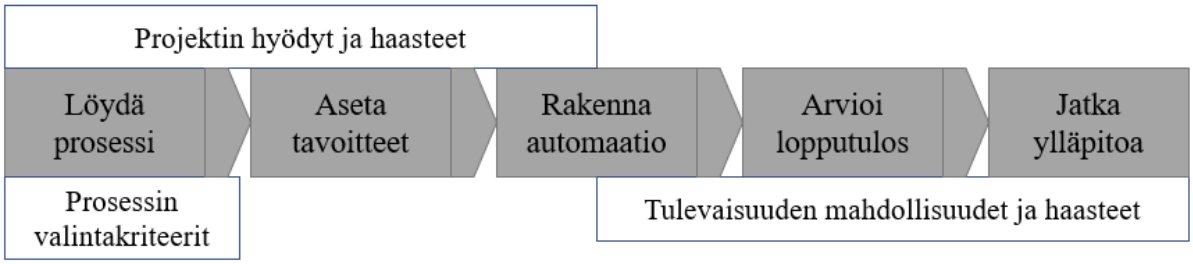
Taulukko 5. Vertailu back-end- ja front-end-automaation välillä (per. Santos et al. 2020)

Osa-alue	Back-end-automaatio	Front-end-automaatio
Liiketoiminnallinen tavoite	Prosessin uudelleensuunnittelu	Olemassa olevan prosessin automaatio
Soveltuvuus	Uusien ohjelmistojen luonti	Olemassa olevien ohjelmistojen käyttö
Integrointimene-telmä	Vuorovaikuttaa liiketoiminnan logiikan sekä tietoihin pääsyä koskevien kerrosten kanssa	Vuorovaikuttaa järjestelmän graafisen käyttöliittymän kautta
Prosessin sopivuus	Sopii parhaiten IT-keskeisiin ja korkean prioriteetin prosesseihin	Sopii parhaiten liiketoimintakeskei-siin ja prosessiasiantuntemusta vaati-viin prosesseihin
Ohjelmoinnin tarve	Vaatii ohjelmointiosaamista	Ei välttämättä vaadi ohjelmointiosaamista
Kehityksen vastuu	Vastuu ohjelmistokehittäjillä	Vastuu liiketoimintafunktioilla
Kehityksen kesto	Pitkä kehitysaika	Lyhyt kehitysaika, kun monimutkai-sia integraatioita ei tarvita

Tässä diplomityössä tullaan hyödyntämään pääasiassa ohjelmistorobotiikkaa sen nopean integraation ja muiden hyötyjen takia, mutta koska yrityksessä ei ole valmiina mitään ohjelmistorobotiikkaohjelmistoa, jonkin verran ohjelmointiosaamista tullaan myös tarvitsemaan.

Automaation toteutukseen liittyy myös jonkin verran haasteita, jotka on otettava huomioon koko projektin ajan. Yksi helposti unohtuva osa missä tahansa ohjelmistoprojektissa on projektin toteutuksen jälkeinen ylläpito ja mahdolliset oikaisutarpeet. Tämä on oleellista eteenkin ohjelmistoroboteissa, sillä ohjelmistojen käyttöliittymät muuttuvat usein ohjelmistopäivitysten myötä. (Kasslin 2017) Ohjelmistorobotit työskentelevät ihmistä nopeammin, mutta siten ne myös tekevät virheitä nopeammin. Jos jotakin yllättävää tapahtuu, esimerkiksi jos internetyhteys on väliaikaisesti katki myrskyn takia, robotti ei välttämättä osaa tarkistaa sitä ja siten suorittaa tehtävänsä väärin tai vain osittain. Poikkeuskäsittelyitä voidaan sisällyttää ohjelmistorobotteihin, mutta kaikkiin katkoksiin ei voida eikä kannattakaan etukäteen varautua, sillä näin työn määrä ja ohjelmistorobotin ylläpitoon menevä aika kasvavat eksponentiaalisesti. (Kirchmer 2017) Ohjelmistoroboteilla on usein suhteellisen vapaa liikkuvuus erilaisten ohjelmistojen ja tietokantojen välillä, joka saattaa luoda tietoturvaongelmia

tai aiheuttaa ongelmia organisaation tietojenkäsittelykäytäntöjen kanssa (Kasslin 2017). Loput haasteista liittyvät viestimiseen ja toimintatapoihin organisaation sisällä. Ohjelmistorobotteihin liittyy paljon väärinkäsityksiä ja harhaluuloja osittain johtuen niiden nimestä, joka liitetään perinteiseen fyysisessä maailmassa toimivaan robotiikkaan, vaikka ohjelmistorobotit nimensä mukaan ovat vain ohjelmia. Toisaalta tämä liitos ohjelmistoihin saattaa johtaa hämmennykseen siitä, kenen vastuulla robottien tulisi olla; IT:n vai niitä käyttävien liiketoimintafunktioiden. (Suri et al. 2017) Monet työntekijät myös mieltävät robotit korvaajina omalle työlleen, mikä ei ohjelmistorobottien paikalla pidä paikkansa, sillä ne eivät voi korvata inhimillistä työtä. Myös organisaatioiden tulisi asennoitua automaatioon työntekijöitä tukevana työkaluna eikä suorana korvaajana. Nämä ongelmat voidaan kitkeä jo projektin alussa hyvällä ja avoimella viestinnällä sekä osallistuttamalla työntekijöitä tai toimihenkilöitä, joiden työtä ollaan automaatiolla tukemassa. (Asatiani & Penttinen 2016) Kuvaan 4 on tiivistetty edellä käytä automaatioprojektin kulkua harmaissa laatikoissa sekä sen eri vaiheissa vaadittavat tiedot valkoisissa laatikoissa.



Kuva 4. Automaatioprojektin kulku ja eri vaiheissa huomioitavat asiat

Automaatioprojektin onnistumisen kannalta siis on oleellista, että viestinnän, muutoshallinnan ja loppukäyttäjien tyytyväisyyden tuloksiin on oltava kohdillaan. Lisäksi korkeamman johdon ymmärrys ja tuki on turvattava sekä selvä päämäärä on oltava määriteltynä, jotta projekti saadaan kunnolla ajettua loppuun. Kun prosessiautomaatio sekä siihen liittyvät käsitteet ovat ymmärryksessä, voidaan siirtyä harkitsemaan sen hyötyjä sekä siihen liittyviä haasteita. Sopivalle automatisoitavalle prosessille on määriteltävä kriteeristö, jonka perusteella sopiva prosessi valitaan. Tämän jälkeen toteutetaan itse automaatiotyö valituilla työkaluilla. Valmista projektia tulisi verrata suunniteltuihin tavoitteisiin, jotta projektin

vaikutus saadaan selville. Lopuksi tulisi pohtia toteutetun projektin tulevaisuuden kehitysmahdollisuuksia sekä myös sen tulevaisuutta uhkaavia haasteita. (Santos et al. 2020) Tämä esitetty konsepti automaatioprojektin toteutukselle muistuttaa huomattavasti prosessijohtamisen elinkaaren kuvamaa operaatiota. Tämä diplomityö pyrki hyödyntämään kummassakin kuvattuja vaiheita. Prosessijohtamisen elinkaaren jaksollisuus ja vaiheet yhdistettynä automaatioprosessin vaiheisiin ja tarvittaviin tietoihin on tämän diplomityön käyttämä teoreettinen malli.

4 Tutkimusprosessin kulku

Tässä luvussa tutustutaan työn kontekstiin esittelemällä ensin lyhyesti työn kohdeyritys eli ANDRITZ Savonlinna Works Oy. Seuraavaksi käydään yleisellä tasolla läpi tutkimusprosessin eri vaiheet, jonka jälkeen tarkastellaan lähemmin, miten työn haastattelut, mallinnus ja sen pohjalta tehty automaatiodemo toteutettiin sekä se, millä logiikalla automatisoitava prosessin osa valittiin. Tämän luvun on tarkoitus toimia alustuksena seuraavalle luvulle, jossa työn tuotoksiin pureudutaan yksityiskohtaisemmin. Tässä luvussa siis kuvataan vain, miten ja mihin perustuen ratkaisut toteutettiin eikä itse tuloksiin oteta vielä juurikaan kantaa.

4.1 ANDRITZ Savonlinna Works Oy:n esittely

ANDRITZ Savonlinna Works Oy (ASWOY) on Savonlinnassa sijaitseva vuonna 1917 perustettu konepaja. Yrityksen tuotanto on erikoistunut sellu- ja paperitehtaiden koneisiin, kuten pesureihin ja painelaitteisiin. Pajan päätuote on niin kutsuttu DD-pesuri (Drum Displacer -pesuri), jota käytetään laajasti selluntuotannon pesuvaiheissa. Yritys tarjoaa myös palveluita muun muassa laitteiden huollon ja asennusten muodossa. ASWOY on osa itävaltalaista ANDRITZ AG -konsernia, jolla on liiketoimintaa sellu- ja paperiteollisuuden lisäksi myös vesivoimassa, metalliteollisuudessa ja erotteluteknologiassa. (ANDRITZ 2023)

ASWOY toimii hyvin kansainvälisillä ja tiukan kilpailun alalla, joten sen on jatkuvasti etsittävä keinoja toimintansa tehostamiseen. Yritys onkin asettanut yhdeksi pidempiaikaiseksi tavoitteekseen toimihenkilötyön tehostamisen ja tätä työtä onkin jo aloitettu muun muassa ohjelmistorobottien testaamisella ja kriittisten prosessien vaiheiden selvittämisellä. Tämän diplomityön tarkoituksena oli jatkaa tätä kehitystyötä rakentamalla toimiva automaatioesittely jostakin yritykselle strategisesta prosessista ja arvioimalla muita suorituskykyä parantavia kehityskohteita näihin strategisiin tavoitteisiin pääsemiseksi.

4.2 Tutkimusprosessin kulku yleisellä tasolla

Tässä diplomityössä haluttiin tutkia toimihenkilötyön suorituskyvyn parantamista yrityksen tarjous-hankintaprosessissa hyödyntäen erityisesti prosessimallinnuksen ja automaation synergiaa. Tutkimusprosessin päävaiheet sekä niihin liittyvät aineistot, menetelmät ja tuotokset on kuvattu Taulukossa 6. Ensimmäisenä vaiheena oli työn teoreettisen viitekehyksen luominen, eli työhön liittyvän teoreettisen aineiston ja aikaisempien julkaisujen läpikäyminen ja synteesi työn teoreettiseksi pohjaksi. Tämä tehtiin kuvailevalla kirjallisuuskatsauksella, jossa tarkasteltiin useita aiheeseen liittyviä vertaisarvioituja lähteitä diplomityössä esitettyjen tulosten pohjaksi. Kuvaileva kirjallisuuskatsaus voidaan jakaa kahteen alatyyppiin, eli narratiiviseen ja integroivaan katsaukseen riippuen siitä, halutaanko jotakin ilmiötä kuvata mahdollisimman suoraviivaisesti vai syvemmin ja monipuolisesti (Salminen 2011). Tässä diplomityössä hyödynnettiin kuvailevaa kirjallisuuskatsausta, sillä sekä prosessijohtaminen että -automaatio ovat erittäin laaja-alaisia aiheita, joita on edullista tarkastella laajemmin useasta lähteestä. Mutta nämä aiheet haluttiin myös kuvata narratiivista tyyliä syvällisemmin ja niistä haluttiin esittää synteesi, joten integroiva tyyli valittiin työn kirjallisuuskatsauksen tyyliksi. Tämän vaiheen tuotokset näkyvät kahdessa edeltävässä luvussa, joissa alustettiin ymmärrys työn keskeisimmistä aiheista, eli prosessijohtamisesta ja -mallinnuksesta käsitteineen sekä prosessiautomaatiosta käsitteineen. Seuraavana vaiheena oli prosessin nykytilan mallintaminen, jossa käytettiin aineistona yrityksen sisäisiä standardeja, vanhoja malleja sekä ohjausryhmän kokouksiin ja asiantuntijoiden haastatteluihin perustuvia muistiinpanoja. Haastattelut käytiin jokaisen tarjous-hankintaprosessin osaprosessin edustajan kanssa hyödyntäen kysymyslistaa, joka löytyy Liitteestä 1. Näillä kysymyksillä saatiin samaan aikaan sekä syvempi ymmärrys prosessin kulusta erilaisista näkökulmista että arvio prosessin eri osien soveltuvuudesta automaatiolle. Tuotoksena tästä vaiheesta saatiin validoitu mallinnus yrityksen tarjous-hankintaprosessista mukaan lukien tiedonkulku siinä. Valmiita malleja analysoimalla ja haastatteluiden pohjalta tehdyistä arvioista löydettiin myös potentiaalisia prosessin osia automaatiodemoa varten, jotka esitellään myöhemmin tässä luvussa. Mallien ja haastatteluiden avulla saatiin myös arviot osaprosessien kriittisimmistä suorituskykyyn vaikuttavista tekijöistä, jotka esitellään tarkemmin seuraavassa luvussa.

Taulukko 6. Tutkimusprosessin kulku

Vaihe	Aineisto	Menetelmä	Tuotos
Työn teoreettisen viite-kehyksen luominen	• Artikkelit • Kirjat • Kokoelmat • Seminaarit	Kuvaileva ja integroiva kirjallisuuskatsaus perustuen vertaisarvioituihin lähteisiin. Erilaisten lähteiden synteesi ja omien hahmotelmien piirtäminen.	• Työssä käytettävien käsitteiden selittäminen • Prosessin valinnan kriteeristö • Prosessijohtamisen elinkaari
Nykytilan mallintaminen ja analyysi	• Yrityksen sisäiset standardit • Vanhat mallit • Kokoukset ohjausryhmän kanssa • Asiantuntijoiden haastattelut	Iteratiivinen BPMN-mallinnus, jossa kokousten ja haastatteluiden kautta saatiin jatkuvasti tarkentuva kuva yrityksen nykytilasta, kunnes saavutettiin haluttu tarkkuus ja asiantuntijoilla validoidut mallit. Mallien analysointi perustuen haastatteluihin ja aikaisemmin määriteltyihin automaatiokriteereihin.	• Todenmukainen kuva tarjous-hankintaprosessin nykytilasta • Tiedonkulku prosessissa • Automatisoitavan prosessin osan löytäminen • Analyysi kriittisimmistä suorituskykyyn vaikuttavista tekijöistä
Automaatiodemon rakennus ja testaus	• Toteutetut mallit • Prosessin tiedonkulku • Automatisoitava prosessin osa • Ostojono ja hankintasuunnitelma • Python-dokumentaatio	Ohjelmoitiin OpenPyXL-kirjastoa hyödyntävä ohjelma, joka luo ja täyttää kolme Excel-tiedostoa, jotka käyttäjä voi ladata SAP:iin automaattisen oston käynnistämiseksi ja hakee tarvittaessa puuttuvan tiedon hankintasuunnitelmasta. Lisäksi ohjelmoitiin PyAutoGUI-kirjastoa hyödyntävä demo ohjelmistorobotista.	• Ohjelmisto, joka muotoilee ja hakee tiedot automaattista ostoa varten • Ohjelmistorobotti, joka demonstroi ohjelmistorobotin nopeutta SAP:in kaltaisessa ympäristössä
Suorituskyvyn arviointi sekä johtopäätökset	• Ohjelmistot ja niiden suoritusajat • Tarjous-hankintaprosessin mallit • Asiantuntijahaastattelut	Arvioidaan suorituskyvyn muutosta perustuen ohjelmien suoritusaikoihin ja muihin hyötyihin. Ehdotetaan jatkokehitysideoita perustuen asiantuntijan mielipiteisiin sekä työn aikana ilmenneisiin omiin havaintoihin. Lopuksi sidotaan havainnot yhteen johtopäätöksiksi.	• Arvio suorituskyvyn muutoksesta • Jatkokehitysehdotuksia • Yleisiä johtopäätöksiä perustuen koko työhön • Miten prosessimallinnus ja -automaatio tukivat toisiaan prosessin aikana?

Seuraavassa vaiheessa suunniteltiin, rakennettiin ja testattiin automaatiodemot perustuen aikaisemmassa vaiheessa määriteltyyn automatisoitavaan prosessin osaan, mallinnukseen ja tiedonkulkuun. Lisäksi hyödynnettiin ohjelman tarvitsemia tietolähteitä, kuten yrityksen ostojonoa ja hankintasuunnitelmaa sekä erilaista dokumentaatiota Python-ohjelmointikielessä ongelmien ilmaantuessa. Tuotoksena tästä vaiheesta syntyi kolme ohjelmaa. Ensimmäinen on laajempi ohjelma, joka OpenPyXL-moduulia hyödyntäen luo ja muokkaa kolme Excel-tiedostoa sellaisiksi, että käyttäjän tarvitsee vain ladata ne SAP:iin automaattisen oston käynnistämiseksi. Toista ohjelmaa kutsutaan, jos tietoa tarvitsee hakea hankintasuunnitelmasta. Kolmas demo-ohjelma on ohjelmistorobotti, joka osoittaa, miten tätä prosessin osaa voitaisiin tehostaa entisestään, jos SAP-ympäristö voitaisiin automatisoida. Ohjelmistorobotti toteutettiin hyödyntäen PyAutoGUI -nimistä Python-moduulia. Viimeisessä vaiheessa kehitetyn järjestelmän luomaa suorituskyvyn muutosta toimihenkilöiden työhön arvioitiin. Tämän sekä asiantuntijan kommenttien ja työn aikana syntyneiden huomioiden pohjalta tehtiin ehdotuksia jatkokehityksestä. Lopuksi pohdittiin koko työn johtopäätöksiä, rajoituksia ja vastauksia johdannossa esitettyihin tutkimuskysymyksiin.

4.3 Haastattelut mallinnuksen pohjana

Suurin osa mallinnuksessa tarvittavista tiedoista saatiin yksittäisiä osaprosessien asiantuntijoita haastattelemalla. Yksittäisiä ja tarkempia haastatteluita käytiin myynti- ja projekti-insinöörin, tuoteinsinöörin, hankinnan suunnittelijan, operatiivisen ostajan sekä toimittajien hallinnan erikoistujan kanssa. Näiden lisäksi pidettiin myös suurempia koko ohjausryhmän sisältäviä kokouksia, joissa pääasiassa käytiin läpi diplomityön edistymistä korkealla tasolla. Suuremmissa kokouksia oli edellisten henkilöiden lisäksi edustajina kehityspäällikkö, laadun ja suunnittelun päälliköt sekä yrityksen toimitusjohtaja, joka toimii myös diplomityön yritysohjaajana. Kokousten edustajat ovat siis vahvasti toimihenkilö- ja tietotyön piirissä työskenteleviä henkilöitä ja siten pystyivät kertomaan funktioidensa toiminnasta suhteellisen tarkasti.

Haastatteluiden pituudet vaihtelivat noin puolesta tunnista reiluun tuntiin. Yleisesti suuremmalla joukolla tehdyt haastattelut venyivät pidemmiksi, kun taas kokoukset pienemmällä joukolla usein pysyivät rakenteellisina ja halutut asiat saatiin selvitettyä nopeammin. Muut

yhteydenotot tapahtuivat pääasiassa sähköpostilla, jolloin epäselvyydet ja tarkemmat kysymykset saatiin selvitettyä nopeasti. Taulukossa 7 on esitelty palaveri kerrallaan niihin osallistuneet henkilöt, niiden kestot sekä niistä saatu tieto.

Taulukko 7. Palaverien kestot, osallistujat ja niistä saatu tieto

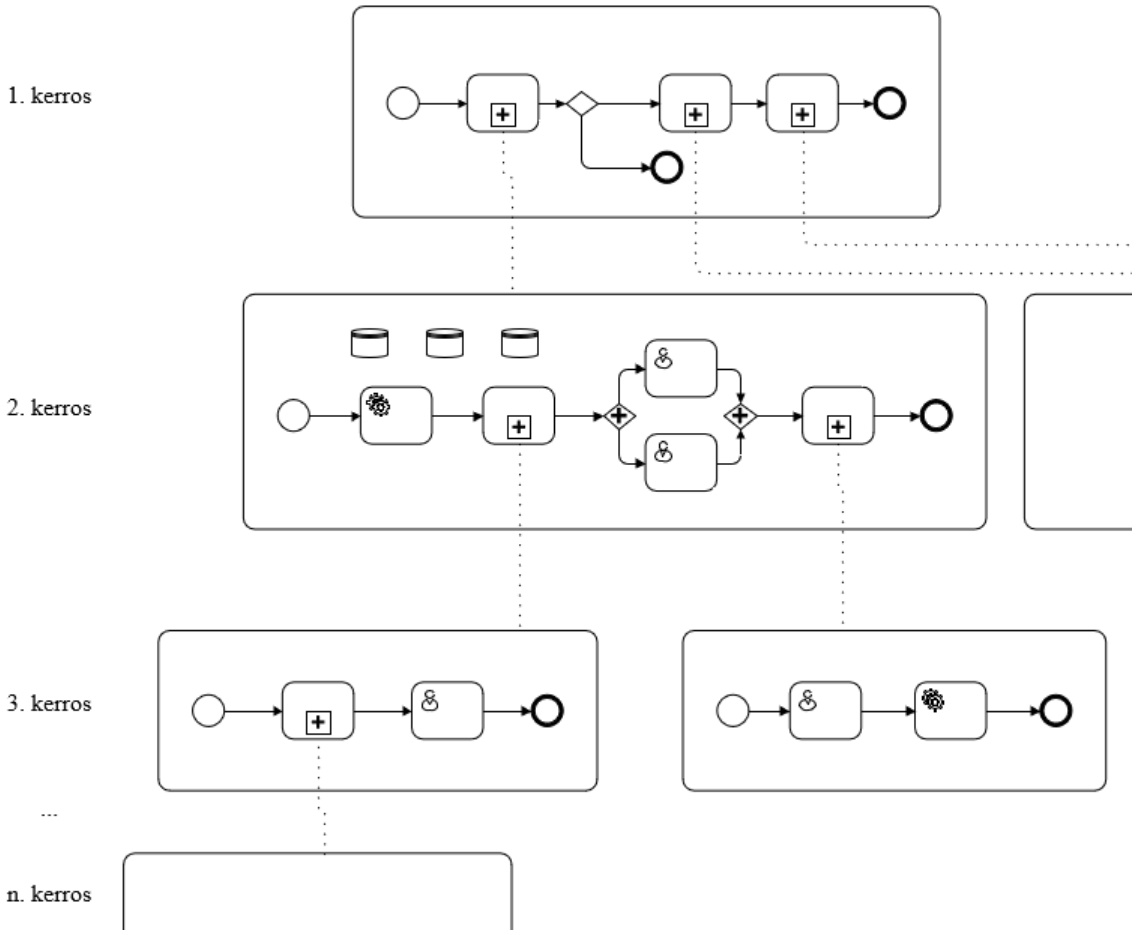
Palaveri	**Kesto (h)**	**Osallistujat**	**Saatu tieto**
Avauspalaveri	~ 1,5	Koko ohjausryhmä	• Yleinen kuva koko tarjous-hankinta-prosessista
Toinen palaveri	~ 1	Koko ohjausryhmä	• Tarkempi kuva tarjous-hankintaprosessista • Keskustelu automaation hyödyntämisestä eri vaiheissa
Yksittäiset palaverit	~ 0,5	Myynti- ja projekti-insinööri	• Tarkka kuvaus jokaisen osaprosessin kulusta • Aloitus- ja lopetuskriteerit, tiedonkulku, resurssit ohjelmistot, poikkeustilat (Liite 1)
	~ 1	Tuoteinsinööri	
	~ 1,5	Hankinnan suunnittelija	
	~ 1,5	Operatiivinen ostaja ja toimittajien hallinnan erikoistuja	
Mallinnuksen jälkeinen palaveri	~0,75	Koko ohjausryhmä	• Katsaus diplomityön nykytilasta • Automatisoitavan prosessin valinta • Suunnitelma lähitulevaisuudelle
Haastattelu automatisoitavan prosessin osan asiantuntijan kanssa	~1	Operatiivinen ostaja ja kehityspäällikkö	• Tarkempi selvitys tiedonkulusta prosessin osassa • Tietolähteiden selvitys

Mallinnusprosessi alkoi pidemmällä yhteisellä kokouksella, jossa selvitettiin prosessin tärkeimpiä vaiheita. Ensimmäisen kokouksen pohjalta tehtiin myös ensimmäiset hahmotelmat korkean tason mallinnuksesta. Toisessa kokouksessa tarkennettiin ensimmäisen kokouksen pohjalta tehty hahmotelma koko prosessista ja keskusteltiin yleisesti automaation soveltuvuudesta prosessin eri osiin, mutta mitään lopullista päätöstä automatisoitavasta osasta ei vielä tehty. Tämän jälkeen aloitettiin yksittäiset haastattelut, joissa hyödynnettiin Liitteestä 1 löytyvää kysymyslistaa. Haastateltavat kävivät tehtävä kerrallaan läpi prosessia sekä kertoivat, mitä ohjelmistoa, tietoa, ohjausmekanismeja ja poikkeuksia mihinkin osaan kuului. Tämän pohjalta pystyttiin tekemään yksityiskohtaisemmat mallit kustakin osaprosessista. Mallien valmistuttua asiantuntijoille lähetettiin niistä kopiot validoitaviksi sekä BPMN-mallinnuksen lukuohjeet. Mallit viimeisteltiin perustuen asiantuntijoiden kommentteihin ja julkaistiin kaikkien ohjausryhmän jäsenten nähtäväksi. Automaatioprosessien potentiaalia pohdittiin myös joidenkin validointikeskusteluiden yhteydessä. Mallien validointien jälkeen käytiin vielä yhteinen palaveri koko ohjausryhmän kanssa, jossa keskusteltiin diplomityön tilasta, jatkosuunnitelmasta ja automatisoitavasta prosessin osasta, joka tässä kohdassa oli jo valittu. Tämän jälkeen pidettiin operatiivisen ostajan ja kehityspäällikön kanssa yhteinen palaveri, jossa selvitettiin automaatiodemon vaatimia tietoja, jotta kehitystyö päästiin aloittamaan.

4.4 Mallinnusprosessin eteneminen vaihe kerrallaan

Haastatteluiden perusteella tehtiin kustakin prosessista kronologinen tehtävälista, jonka perusteella prosessien rungot pystyttiin rakentamaan. BPMN-mallinnuksen runko perustuu tehtäviin (task) ja osaprosesseihin (subprocess), jotka sisältävät useampia tehtäviä tai jopa uusia osaprosesseja. Tässä mallinnuksessa hyödynnettiin vahvasti osaprosessien hierarkkista linkitystä, jotta mallit saatiin pysymään järkevän kokoisina ja helpommin luettavina. Hierarkkisessa mallinnuksessa yksittäiset mallit voidaan usein mahduttaa pieneenkin tilaan, kuten yhdelle sivulle, mutta se voi uhrata mallin luettavuutta, kun lukijan on jatkuvasti hypittävä eri sivujen välillä. Tämän takia mallinnuksessa on hyvä pitää tasapaino eri tasojen ja yksittäisten mallien laajuuksien välillä. Periaatteessa notaation syntaksin kannalta sillä ei ole väliä, kuinka monta kerrosta mallissa on tai kuinka suuri yksi yksittäinen malli on, vaan nämä päätökset ovat suurelta osin mallintajan päätettävissä. Tämän työn malleissa on hyödynnetty Kuvassa 5 kuvattua rakennetta, jossa ylimmällä tasolla on vain osaprosesseja, koko

prosessin laukaisija ja mahdolliset lopputilanteet. Jokaisesta osaprosessista lähtee linkki tarkempaan malliin, joka on aina yhden kerroksen alempana verrattuna sen lähteeseen.



Kuva 5. Diplomityön hierarkkisen mallinnuksen logiikka

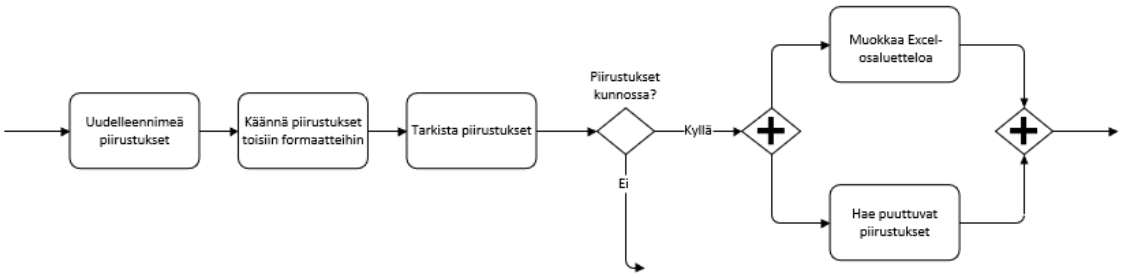
Pääosat tämän diplomityön malleissa ovat ylin kokonaiskuva (1. kerros Kuva 5), sitä alemmat laajemmat mallit (2. kerros Kuva 5) sekä niiden alapuolella olevat pienemmät mallit (3.-n. kerros Kuva 5). Ylimmän kerroksen yleiskuvalle oleellisimmat osat ovat prosessin laukaisu, lopputilat sekä tärkeimmät osaprosessit ja päätökset. Ylimmän tason mallin tulisi olla nopeasti luettavissa, mahduttava yhdelle sivulle ja jokaisen osaprosessin tulisi linkittyä tarkempaan malliin. Ylimmän tason alla ovat tarkimmat mallit koko prosessin päätehtävistä. Nämä mallit ovat kaikkein laajimpia ja ne sisältävät kaikki osaprosessin käyttämät ohjelmistot ja tiedonkulun. Nämä mallit koostuvat suurimmaksi osaksi tavallisista tehtävistä, mutta

jotkin laajemmat tehtävät on suljettu aliprosesseiksi, jotka taas linkittyvät alempien kerrosten osaprosesseihin. Näitä alempia kerroksia voi periaatteessa olla rajattoman paljon, mutta mallin luettavuus ja mallille sopiva tarkkuustaso on aina pidettävä mielessä.

Seuraavaksi tarkastellaan pätkää hankinnan suunnittelun osaprosessista ja esitetään sen avulla, miten mallit tässä työssä muodostettiin. Esimerkkinä käytetty prosessin osa kuvaa piirustusten käsittelyä ja osaluettelon täyttöä. Ensin tulisi tietää, mikä tai mitkä asiat voivat laukaista prosessin ja mitä lopputiloja prosessilla voi olla. Tämä auttaa hahmottamaan prosessin mittakaavan, suhteen sitä seuraaviin prosesseihin ja ne myös auttavat rajaamaan prosessiin kuuluvat tehtävät. Seuraavaksi tehtävät on hahmoteltu aikajärjestykseen Kuvan 6 osoittamalla tavalla perustuen haastattelussa koottuun listaan ja muihin muistiinpanoihin. Tehtävien sisällöt on myös muokattu käskymuotoisiksi, sillä se kuuluu BPMN-syntaksin muotosääntöihin.

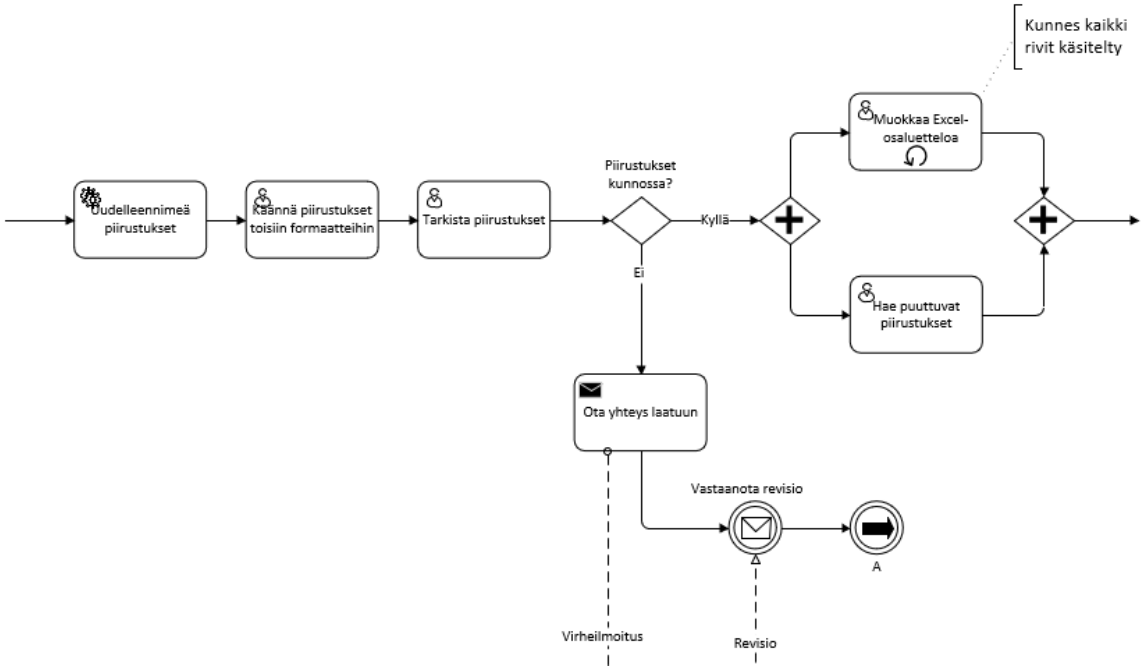
Kuva 6. Mallin tehtävien kronologinen hahmotelma

Tässä vaiheessa malli on vielä hahmotelma ja siitä puuttuu käytännössä kaikki logiikka ja konteksti. järjestysvuo (engl. sequence flow) kuvataan yhtenäisillä nuolilla BPMN-notaatiossa ja päätöksentekokohtia sekä järjestysvuon jakautumista kuvataan salmiakin muotoisilla logiikkaporteilla (engl. logic gateway). Porttien kohdalla tehdään usein binäärinen järjestysvuon jako, mutta myös ehdollinen jako useammalle polulle tai järjestysvuon kulun jakaminen rinnakkaisiin tehtäviin ovat mahdollisia. Kuvassa 7 esimerkkiprosessiin on lisätty järjestysvuon kulku sekä kaksi erilaista logiikkaporttia.



Kuva 7. Malli järjestysvuolla ja logiikkaporteilla

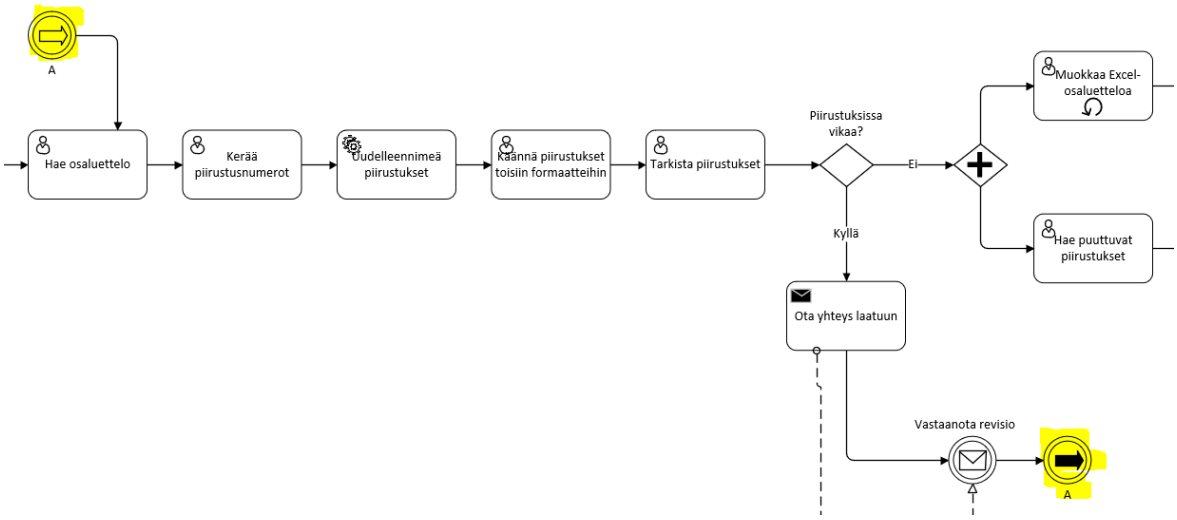
Tyhjä logiikkaportti on niin kutsuttu ehdoton TAI-portti (engl. XOR gate), joka tarkoittaa sitä, että järjestysvuo voi jatkua vain toisella polulla riippuen testin tuloksesta, eli tässä tapauksessa siitä, ovatko piirustukset kunnossa. Tässä kohdassa vain myönteinen polku on mallinnettu, mikä on yleinen mallinnustapa. Poikkeukset ja niiden käsittelyt lisätään yleensä vasta, kun koko mallin ’perustilanne’ on valmis. Haastateltavat eivät aina myöskään muista kertoa poikkeustilanteiden synnyistä tai käsittelyistä, jos ne ovat harvinaisia, jolloin mallintajan on selvitettävä niitä jälkeenpäin tai oltava aktiivinen niiden kyselyssä haastattelun aikana. Toinen kuvassa esiintyvä logiikkaportti on niin kutsuttu JA-portti (engl. AND gate), joka jakaa järjestysvuon kokonaan tai osittain rinnakkain suoritettaviin osiin. Järjestysvuot yhdistyvät toiseen JA-porttiin, jolloin prosessi jatkuu taas yhtenä virtana kummankin tehtävän valmistuttua. Kuvassa 8 esitetyssä vaiheessa on lisätty sekä TAI-portin toinen polku että tehtävien tyypit ja lisäksi yksi silmukkatehtävä.



Kuva 8. Malli tehtävien roolien, silmukoiden sekä viestivirtojen kanssa

Kaikki prosessin tehtävät eivät ole luonteeltaan samanlaisia ja sen merkitsemiseksi käytetään BPMN-notaatiossa pieniä kuvakkeita tehtävälaatikoiden vasemmassa yläkulmassa. Tehtävä ’Uudelleennimeä piirustukset’ on automatisoitu, joten siinä on pieni hammaspyörämerkki, kun taas esimerkiksi tehtävässä ’Tarkista piirustukset’ tarvitaan inhimillistä harkintaa, joten se on ihmisen manuaalisesti tekemä työ, jolloin siinä on pieni henkilön kuva. Tämä erottelu tarkentaa prosessin kulkua ja on myös avuksi, jos halutaan löytää potentiaalisia tehostustoimia tai paikkoja automatisaatiolle. Kirjesymbolia käytetään tehtävissä, joissa käsitellään tiedon lähettämistä tai vastaanottoa. Väritetty kirjekuori tarkoittaa viestin lähettämistä (engl. throw), kun taas väritön kirjekuori tarkoittaa viestin vastaanottamista (engl. catch). Mallikuvassa näkyy yksi lähetystehtävä, josta lähtee katkoviivallinen nuoli ja vastaanottotehtävä, johon saapuu katkoviivallinen nuoli. Tiedon ja viestien kulun selvittäminen prosessin sisällä on tärkeää, sillä vaikka ne vaikuttavat yksinkertaisilta ja arkipäiväisiltä tehtäviltä, todellisuudessa ne ovat yksiä kriittisimpiä tekijöitä siinä, onnistuuko prosessi ilman poikkeuksia tai viivästyksiä. Tämän työn haastatteluissa kävi ilmi useammassa prosessin osassa, että tiedon saanti oikeaan aikaan ja täydellisessä muodossa ovat usein tärkeimpiä tekijöitä prosessin suorituskyvyn kannalta. Tämä ei tietysti ole kovin suuri yllätys, sillä tiedon laatu korostuu asiantuntija- ja tiedonkäsittelytyössä suuresti.

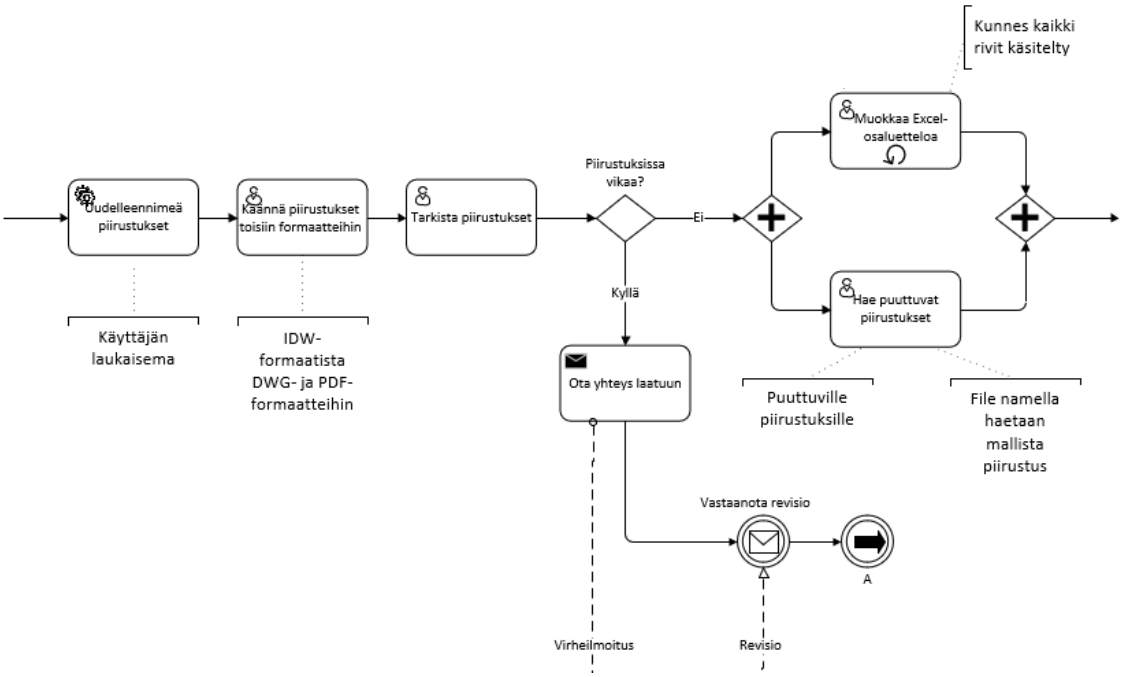
Kuvassa nähdään myös silmukkatehtävä ja siihen liittyvä ehto irrallisessa kommenttikentässä, joka on liitetty katkoviivalla tehtävään. Tässä tapauksessa tehtävä sisältää tiettyjen tietojen muokkaamista ja lisäämistä kaikille Excel-riveille. Silmukkatehtäviä käytetään, kun halutaan kuvata iteratiivisia tehtäviä tai aliprosesseja. Esimerkkimallin tapaus soveltuu erityisen hyvin silmukkatehtävälle, sillä toistojen määrä vaihtelee tapauksesta riippuen, koska puuttuvien rivien määrä ei ole vakio. Tehtävän ’Vastaanota revisio’ jälkeen nähdään ympyrän sisällä nuoli, johon järjestysvuo näyttää pysähtyvän. Tämä niin kutsuttu linkkitapahtuma (engl. link event) on kuitenkin osa suurempaa silmukkaa, joka lähettää järjestysvuon kulun aikaisempaan kohtaan prosessia, joka voidaan nähdä Kuvassa 9.



Kuva 9. Osaprosessin linkkitapahtuman kumpikin puoli

Kuten viestitapahtumissakin, värjätty nuoli lähettää järjestysvuon ja värjäämätön nuoli ottaa sen vastaan. Linkkitapahtumat ovat vain visuaalinen apu, sillä tehtävät ’Vastaanota revisio’ ja ’Hae osaluettelo’ olisi myös voitu yhdistää tavallisella järjestysvuota kuvaavalla nuolella. Tämä esimerkki kuitenkin demonstroi muutamaa oleellista asiaa BPMN-mallinnuksessa. Ensinnäkin se, että silmukoita voidaan rakentaa monella erilaisella tavalla tilanteesta riippuen ja toisekseen se, että BPMN-mallin vahvuuksiin kuuluu sen visuaalinen selvyys ja monimuotoisuus. Samanlaisen mallin voi rakentaa monella eri tavalla ja ne voivat kaikki olla

notaation syntaksin kannalta oikein. BPMN-mallinnukseen, kuten myös muuhunkin mallinnukseen tai vaikka ohjelmointiin, kuuluu paljon mallintajan itsensä tekemiä pieniä päätöksiä, jotka lopulta johtavat lopulliseen haluttuun malliin, mutta se miten lopputulokseen päästiin voi erota huomattavastikin jonkun toisen vastaavasta työstä. Kuvassa 10 on kuvattu viimeistelty osaprosessi.



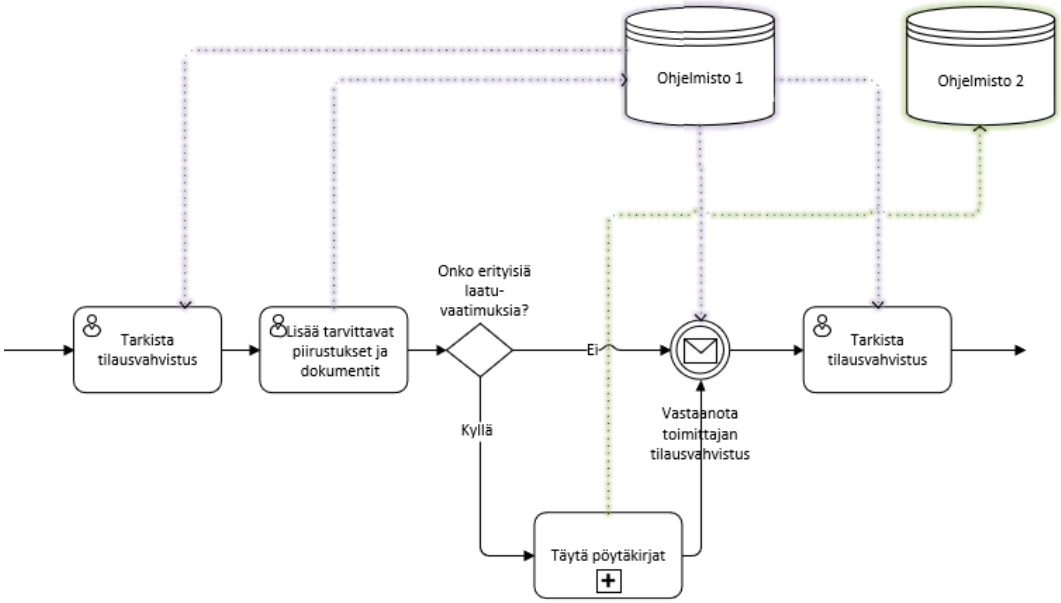
Kuva 10. Kommenteilla viimeistelty osaprosessi

Mallinnus usein viimeistellään tarkentavilla kommenteilla. Kommentit merkitään samalla tavalla kuin silmukkatehtävän ehtolause ja ne usein jätetään omalle kerrokselleen mallissa, jolloin ne voidaan helposti piilottaa mallin lukemisen selventämiseksi. Mallissa tehtävät kuvataan usein vain muutamalla sanalla, jolloin niiden tarkoitus voi jäädä hämäräksi lukijalle. Esimerkiksi tehtävän ’Käännä piirustukset toisiin formaatteihin’ nimestä ei selviä, mihin formaatteihin piirustukset tulisi kääntää tai missä formaatissa ne alun perin ovat. Lyhyellä kommentilla tällaiset tehtävät voidaan selventää mallissa, mutta yksi oleellisen tärkeä osa mallinnusta on myös sanallisen selityksen kirjoittaminen mallin rinnalle. Tässä selityksessä tehtävien yksityiskohtia voidaan avata kommenttejakin syvemmin. Kirjallinen selite on

myös mallin jakelun kannalta hyvä olla olemassa, sillä vaikka BPMN-mallinnus on suhteellisen helppolukuinen jokaiselle vuokaavioiden kanssa työskennellyille henkilölle, on kirjallinen selitys prosessin kulusta aina selkeämpi ja vähemmän tulkinnanvarainen esitystapa suurimmalle osalle lukijoista.

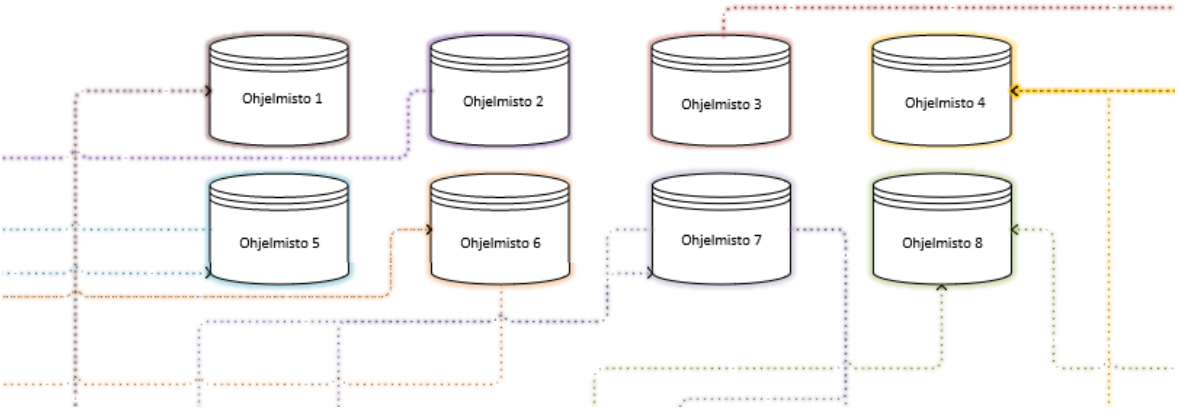
4.5 Tiedonkulku ja ohjelmistot mallinnuksessa

Tiedonkulku on erittäin tärkeä osa prosessia eteenkin nykyään, kun tietotyö ja erilaisten ohjelmistojen hyödyntäminen työn teossa ovat yleistyneet. Varsinkin isommissa yrityksissä olisi hyvä kartoittaa tasaisin väliajoin, mitkä ohjelmistot ovat oikeasti käytössä ja mitkä ovat niin sanotusti jääneet kummittelemaan. Silti ohjelmistot ja tiedonkulku unohtuvat usein mallinnuksessa todennäköisesti siksi, että mallinnus on perinteisesti keskittynyt konkreettisten tehtävien kuvaamiseen tietotyön sijaan. Poikkeuksena ovat erilaiset IT-prosesseja kuvaavat mallit, jotka käsittelevät käytännössä pelkästään tiedon kulkua prosessin aikana. Tässä diplomityössä yhdistettiin tiedonkulun mallinnus ja konkreettinen prosessin kulku. Tämä toteutettiin käyttäen BPMN 2.0 -syntaksin tietopankkikuviota sekä niistä lähtevien ja niihin saapuvia tietoassosiaationuolia. Tästä voidaan nähdä esimerkki Kuvassa 11.



Kuva 11. Esimerkki ohjelmistojen ja tiedonkulun esittämisestä osana prosessia

Kuvan 11 esittämä prosessin osa on leikattu operatiivista ostoa kuvaavasta osaprosessista. Todellisuudessa operatiivisessa ostossa hyödynnetään kahdeksaa erilaista ohjelmistoa ja kuvan kaksi ohjelmistoa liittyvät myös muihin tehtäviin, mutta esimerkin selvyyden takia ne on jätetty pois kuvasta. Tässä prosessin osassa hyödynnetään kahta ohjelmistoa, joiden toiminta yleisellä tasolla liittyy tietojenhallintaan ja kommunikaatioon. Tehtäviä ja tietovarastoja yhdistävät pilkulliset yksisuuntaiset nuolet, jotka edustavat tiedon kulkua. Esimerkiksi Kuvan 10 ensimmäinen tehtävä ’Tarkista tilausvahvistus’ vastaanottaa nuolen ensimmäisestä tietovarastosta. Tämä tarkoittaa tiedon lukemista tehtävässä, sillä toimittajien tilausvahvistukset saapuvat tähän tietovarastoon. Vastakkaisesti mallin seuraavassa tehtävässä ’Lisää tarvittavat piirustukset ja dokumentit’ nuoli lähtee tehtävästä ja päättyy ensimmäiseen tietovarastoon, mikä tässä tapauksessa tarkoittaa kirjoittamista, sillä mahdolliset piirustukset ja muut dokumentit lähetetään tämän ohjelmiston kautta toimittajalle. Jos tilauksessa on erillisiä laatuvaatimuksia, siitä on täytettävä tarvittavat pöytäkirjat, mikä on merkitty kirjoitusnuolella kuvan toiseen tietovarastoon. Toimittajan tilausvahvistuksen vastaanottaminen taas on merkitty ensimmäisestä tietovarastosta lähtevällä nuolella, mikä merkitsee tiedon lukemista ja niin edelleen. Kaikki tiedonkulku malleissa on merkitty käyttäen samaa logiikkaa.



Kuva 12. Tietovarastojen ja tiedonkulkua kuvaavien nuolten värikoodaus

Lukemisen helpottamiseksi ohjelmistot ja tiedonkulkua kuvaavat nuolet on asetettu omalle kerrokselleen, jolloin ne voidaan piilottaa helposti näkyvistä. Ohjelmistot ja

tiedonkulkunuolet on myös värikoodattu riippuen ohjelmistosta nuolien erottelun helpottamiseksi (Kuva 12).

4.6 Automatisoitavan prosessin osan valinta

Tässä luvussa käsitellään malleihin perustuvia mahdollisia automaatiokohteita. Kohteet valittiin käyttäen aiemmin koottua automaatiokriteeristöä (Taulukko 4) sekä prosessin osia ympäröivää kontekstia, joka selvitettiin mallinnuksen yhteydessä tehtyjen haastatteluiden pohjalta. Automaatiokohteita löytyi muun muassa hankinnan alkusuunnittelusta, hankinnan suunnittelusta ja automaattisesta ostosta. Seuraavaksi esitellään ne osa-alueet, joita harkittiin tähän diplomityöhön, tosin muitakin kohteita löydettiin.

4.6.1 Hankinnan alkusuunnittelun nimiketietojen hakeminen tietokannasta

Tarjous-hankintaprosessin alussa, kun asiakas on tehnyt tilauksen ja kun projektin alkutoimenpiteet on suoritettu, alkaa hankinnan alkusuunnittelussa nimikelistan kokoaminen. Tälle prosessin osalle tehtävä automaatio vertaisi hankinnan nimikelistaa tietokannassa olemassa oleviin nimiketietoihin, jolloin hankittavien nimikkeiden mahdolliset puutteelliset tiedot voitaisiin täyttää automaattisesti odotusajan vähentämiseksi. Tällä hetkellä puuttuvat tiedot haetaan ja syötetään käsin tai niiden saapumista odotetaan asiakkaalta, jonka jälkeen tiedot syötetään vielä toiseen tietokantaan. Automaatio nopeuttaisi nimikelistan kokoamista sekä hyödyntäisi paremmin jo kerättyä ja varastoitua tietoa erilaisista nimikkeistä. Haasteena on tietokannan konfiguraattorin suhteellinen monimutkaisuus ja tiedon keräämiseen vaadittava tietomäärä. Tässä projektissa kuitenkin vältyttäisiin kosketuspinnasta SAP:iin, mikä tekisi ratkaisusta helpommin toteutettavan ja pitkäikäisemmän.

Tämän projektin toteuttaminen olisi strategista, sillä se parantaisi toimihenkilötyön suorituskykyä ja olemassa olevan datan ja tiedon hyödyntämistä. Prosessin osa soveltuisi hyvin automaatiolle, sillä se koostuu suurimmaksi osaksi tiedon hakemisesta eri rajapinnoista. Prosessin osa ei ole arkipäiväinen, mutta nimikelistat ovat pitkiä, joten työn toistuvuus on korkea. Haasteena tämän prosessin toteutuksessa on siinä harkittavien muuttujien määrä ja epäsäännöllisyys. Vastaavia nimikkeitä ei voida vain hakea tietyllä nimellä, sillä

konfiguraattoriin tallennetut nimikkeet saavat aina uuden uniikin nimen. Nimikkeiden kaikkia ominaisuuksia on siis verrattava kaikkiin vastaavien nimikkeiden ominaisuuksiin ja niiden on täsmättävä täydellisesti, jotta tallennetun nimikkeen tietoja voidaan hyödyntää. Tämä on erittäin työläs tapa automatisoida, mutta teknisesti se on mahdollista toteuttaa. Prosessin osa voidaan rajata suhteellisen tarkasti eikä se vaadi inhimillistä harkintaa lopputarkastusta lukuun ottamatta. Datan ja tiedon jäsentely on vaihtelevaa ja joskus tiedot jäävät vajaiksi asiakkaan hitaan ja puutteellisen viestinnän takia tai koska nimikkeestä ei ole olemassa historiatietoa. Kuitenkin tiedossa olevat nimikkeet voidaan verrata tietokantaan, mikä saattaisi nopeuttaa prosessin siirtämistä hankinnan suunnittelun vaiheeseen. Prosessi on myös suhteellisen muuttumaton eikä SAP-ympäristön muutoksesta koituisi suoria seurauksia ratkaisulle.

4.6.2 Hankinnan suunnittelun Excel-osaluettelon valmistelu

Hankinnan suunnittelussa käytetään apuna Excel-tiedostoa, johon tarvittavat toimittaja- ja muut tiedot kerätään. Vaikka se mallissa on kuvattuna vain yksittäisenä tehtävänä, se voi viedä huomattavan paljon toimihenkilön työaikaa, jos kyseessä on suurempi tilaus. Excelin kokoamiseen liittyy joitain säännönmukaisuuksia, jotka voi tehdä automatisoidulla logiikalla, mutta suurin osa haetuista tiedosta vaatii inhimillistä harkintaa. Automaation tarkoituksena tässä tapauksessa olisi ihmistyön helpottaminen ja varsinaiseen ajattelutyöhön nopeammin pääseminen. Ratkaisu vaatisi tiedonhaun useista rajapinnoista, SAP mukaan lukien, joka luo omat haasteensa. Excelin muokkaaminen ja muiden käsittelyiden tekeminen automaattisesti on kuitenkin suhteellisen yksinkertaista.

Tämän prosessin osan automaatio parantaisi hankinnan suunnittelun suorituskykyä, jolloin se olisi linjassa yrityksen strategian kanssa. Projekti on suhteellisen helposti toteutettavissa, sillä tiedon haku, ja Excel-tiedostojen käsittely soveltuvat hyvin automaatiolle. Prosessin osa ei ole arkipäiväinen, mutta se on toistuva, sillä osaluettelossa on suuri määrä rivejä vähänkin suuremmille tilauksille. Merkinnät Excel-tiedostoon tehdään säännönmukaisin perustein, mutta projektiin mukaan otettavien sääntöjen rajaus ei ole välttämättä itsestään selvää. Suurin osa prosessin päätöksistä vaatii inhimillistä harkintaa, joten koko prosessin osaa ei missään tapauksessa voida automatisoida kokonaan. Tämä on kuitenkin kriittinen ja

palveluaikaa paljon vievä hankinnan suunnittelun osa, joten osittainenkin automaatio todennäköisesti näkyisi osaprosessin suorituskyvyssä. Prosessiin liittyy jäsennellyn tiedon siirtämistä ohjelmistojen välillä sekä Excel-tiedoston muokkaamista, mitkä tekevät siitä otollisen projektin automaatiolle. Prosessin osa on suhteellisen muuttumaton, mutta jos esimerkiksi toimittajien vaatimat tiedot muuttuvat tai jos yritys kokeilee uusia toimittajia, tulee näiden muutosten lisääminen huomioida ratkaisun ylläpidossa.

4.6.3 Hankinnan suunnittelun viimeiset SAP-merkinnät

Hankinnan suunnittelun osaprosessin loppuvaiheilla tehdään useita viimeisteleviä merkintöjä työnumeroille ennen kuin ne siirretään niitä vastaaville nimikkeille. Sen jälkeen kirjataan mahdolliset ajoitukset, ostotekstit, laatuvaateet sekä kirjanpitotili ennen kuin lopullinen vapautus tehdään ostajille. Nämä ovat suhteellisen rutiininomaisia, yksinkertaisia ja toistuvia tehtäviä, jolloin ne soveltuvat automaatiolle paljon paremmin kuin ihmisille käsin tehtäväksi. Haasteena tämän projektin toteutukselle on se, että kaikki nämä vaiheet on tehtävä SAP-ympäristössä, vaikkakin teknisesti projekti olisi suhteellisen yksinkertainen toteuttaa esimerkiksi ohjelmistorobotilla.

Osaprosessin suorituskyky todennäköisesti paranisi, jolloin se on yrityksen strategian mukainen, mutta ennen automaatioprojektin toteutusta tulisi selvittää, onko automaatioon käytetty aika ja vaiva siitä saatujen hyötyjen arvoista. Käsin tehtynä prosessin osan toteutuksessa ei mene hirveän pitkään, mutta sen automaatio on teknisesti suhteellisen helposti toteutettavissa ja vähentäisi turhaa toistoa. Tämä prosessin osa ei ole arkipäiväinen, sillä se tehdään vasta lopuksi, kun koko osaluettelo on kasassa, mutta siihen kuuluu paljon peräkkäisiä pieniä tehtäviä, jotka on tehtävä joka kerta ennen suunnitelman lopullista vapauttamista operatiivisille ostajille. Prosessin osan eri vaiheet ovat säännönmukaisia, eivätkä ne vaadi inhimillistä harkintaa ja ne voidaan helposti rajata tiettyyn ympäristöön. Vaikka prosessin sisäinen data ja tieto on jäsenneltyä, prosessi ei juurikaan vaadi tiedon siirtämistä ja kaikki toiminnot tehdään yhden ohjelmiston sisällä. Koska prosessi suoritetaan kokonaan SAP-ympäristössä, se ei ole täysin muuttumaton tulevan versiomuutoksen takia.

4.6.4 Puuttuvien tietojen korjaaminen automaattisen oston mahdollistamiseksi

Yrityksellä on käytössä automaattinen ostotyökalu operatiivisen oston yhteydessä tietyille nimikkeille. Sen käyttöönottaminen uusille nimikkeille vaatii kuitenkin suuren tietomäärän syöttämisen jokaiselle nimikkeelle, johon ostajalla on harvoin aikaa ostojonon läpikäynnin aikana, jolloin nämä nimikkeet joudutaan ostamaan käsin. Järjestelmään kuuluvista nimikkeistä voi myös syystä tai toisesta puuttua vaadittavaa tietoa, jolloin sen kanssa on meneteltävä samalla tavalla kuin uuden nimikkeen kanssa. Tämän prosessin automatisaatio nostaisi siis jo olemassa olevan automatisaation käyttöastetta ja nopeuttaisi ostajan työtä. Tämä myös loisi kaskadiefektin, kun enemmän nimikkeitä löytyy jo automaatio-oston piiristä, jolloin yhdelläkin ajolla olisi suuri merkitys tulevaisuudessa. Haasteena tälle projektille on se, että suurin osa tiedoista on syötettävä SAP-rajapinnan kautta, jonka automaatio on tällä hetkellä epäselvää voimassa olevien sopimusten takia.

Tämän projektin toteuttaminen olisi strategisesti merkittävä, sillä se lisäisi automatisaation määrää sekä tehostaisi toimihenkilötyötä, jotka ovat molemmat organisaation strategisia tavoitteita. Koko operatiivisen oston suorituskyky paranisi, jos osa käsin tehtävistä ostoista voitaisiin automatisoida tai jos ostajan ei tarvitse täyttää vaadittavia tietoja käsin. Tämä automaatioprojekti on suhteellisen helposti toteutettavissa, sillä se koostuu vain historiatiedon hakemisesta ja syöttämisestä eri rajapintojen läpi, mikä on suhteellisen yksinkertaista toteuttaa ohjelmallisesti. Tämä prosessin osa on ostajalle suhteellisen arkipäiväinen ja se toistuu usein ostojonoa käsiteltäessä, joten se sopii hyvin automaatiolle. Prosessin osan voidaan sanoa olevan säännönmukainen, sillä sen tarvitsemat tiedot ovat aina samanlaiset ja ne löytyvät vakiopaikoilta ja täten prosessin osan suoritus ei myöskään tarvitse inhimillistä harkintaa. Projekti voidaan myös rajata suhteellisen tarkasti ja sillä on selkeä alku- ja lopputila. Kuten aikaisemmin todettiin, tämän prosessin toteutus soveltuu teknisesti hyvin automaatiolle, sillä siinä käytännössä vain siirrellään jäsenneltyä tietoa eri rajapintojen välillä. Prosessi ei ole täysin muuttumaton, sillä yrityksen käyttämä SAP-ympäristö tulee päivittymään tämän vuoden aikana. Tässä kohdassa on kuitenkin vaikea sanoa, kuinka paljon ympäristö tulee muuttumaan ja miten paljon muutos vaikuttaa tiedonsyöttöön. On hyvä kuitenkin huomioida, että automaatio-ohjelmien ja ohjelmistorobottien ylläpidon perustoimiin kuuluu ohjelmistopäivitysten huomioiminen ja robottien uudelleenkalibroiminen päivitysten jälkeen.

Taulukkoon 8 on kerätty sekä automatisaatioon harkittavat prosessien osat että automatisoitavan prosessin valintakriteerit. Valintakriteerit käsiteltiin aikaisemmin luvussa ”3.3 Automatisoitavan prosessin valintakriteerien asetus”, jossa ne tiivistettiin Taulukkoon 4. Edeltävien kuvausten perusteella Taulukkoon 8 on merkitty ruksilla, jos prosessin osa täyttää kriteerin.

Taulukko 8. Vertailu siitä, miten hyvin eri vaihtoehdot täyttävät asetetut automaatiokriteerit

Kriteerit	**Nimikelistan kokoaminen**	**Hankinnan suunnittelun Excel**	**Hankinnan suunnittelun SAP-merkinnät**	**Korjaa puuttuvat tiedot**
Strateginen merkityksellisyys	X	X	X	X
Prosessin suorituskyky	X	X	X	X
Toteutettavuus	X	X	X	X
Toistuvuus	X	X	X	X
Arkipäiväisyys				X
Säännönmukaisuus (standardisoitu)		X	X	X
Tarkka rajattavuus	X		X	X
Yksiselitteiset päätökset (ei inhimillistä harkintaa)	X		X	X
Jäsennelty data/tieto		X	X	X
Datan/tiedon siirtäminen	X	X		X
Useiden ohjelmistojen käyttö	X	X		X
Prosessin muuttumattomuus	X	X		

Kaikki harkitut osaprosessit ovat varteenotettavia, sillä kuten voidaan nähdä Taulukosta 8, kaikki harkitut prosessin osat täyttävät suurimman osan kriteereistä. Näistä tehtäväksi valittiin puuttuvien tietojen korjausten automatisointi, sillä prosessin osan katsottiin täyttävän automaatiokriteerit parhaiten, sen katsottiin olevan laajuudeltaan sopiva diplomityöhön ja sen katsottiin tuovan eniten kvantitatiivisesti osoitettavaa hyötyä suhteessa sen toteutukseen kuluvaan aikaan.

4.7 Automaatioratkaisun toteutus

Automaatioprojektin toteuttamiseksi tarvitaan suunnitelma, jotta päästään hallitusti haluttuun määränpäähän. Automaatioprojektin kulkua on jo avattu tässä työssä luvussa ”3.4 Automaatioprojektin toteutus” ja tiivistetty Kuvaan 4. Ensimmäisenä vaiheena oli löytää automatisoitava prosessi perustuen valintakriteereihin sekä pohdittava projektin hyötyjä ja haasteita. Tämä vaihe tehtiin edeltävässä luvussa. Seuraavaksi tuli asettaa projektin tavoitteet, jotka jaettiin lyhyen ja pitkän aikavälin tavoitteisiin. Tässä diplomityössä pyrittiin suorittamaan asetetut lyhyen aikavälin tavoitteet diplomityön suhteellisen lyhyen työstöajan vuoksi. Nämä tavoitteet olivat toimivan automaatiodemonstraation kehittäminen, poikkeusten käsittely oleellisimmille poikkeustiloille ja mitattava parannus ostajan suorituskykyyn. Myös pidemmän aikavälin tavoitteita asetettiin perustuen automaation toimintaan sekä prosessiasiantuntijan omiin ehdotuksiin. Tavoitteet ovat seuraavat: automaattisesti tehtyjen tilausten määrän kasvu ja automaattisen oston piiriin kuuluvien nimikkeiden laajentaminen useampiin toimittajiin. Lyhyen aikavälin tavoitteet olivat analyyttisempia ja pyrkivät valaisemaan yrityksen kypsyyttä toimihenkilötyön automaatiolle. Pidemmän aikavälin tavoitteet taas ovat määrällisempiä ja pyrkivät ratkaisujen laajentamiseen.

Työssä päädyttiin automatisoimaan ostonfunktioon kuuluva prosessin osa, jossa tietoa haetaan erilaisista historiatiedon lähteistä ja kirjoitetaan erilaisiin SAP-transaktioihin, jotta automaattinen tilaus menisi läpi. Itse tilauksen automaatio on jo olemassa ja koettu yrityksessä erittäin hyödylliseksi, mutta pullonkaulana sen laajemmalle käytölle on ollut ostettavan nimikkeen tiedonhaun hitaus. Ostajat joutuvat ostojonoa läpikäydessä arvioimaan, ehtivätkö he täyttämään puuttuvan tiedon vai tekevätkö tilauksen vain käsin, ja he usein päätyvät kiireen takia tekemään ne käsin. Myös tilanteessa, jossa heillä on aikaa täyttää puuttuvat tiedot,

palveluaikaa kuluu turhan paljon, kun tietoa on haettava ja syötettävä manuaalisesti käsin. Tiedon nopea automaattinen kerääminen, koonti ja syöttäminen nostaisivat jo olemassa olevan automaatioratkaisun käyttöastetta, säästäisivät ostajien aikaa ja vähentäisivät virheitä tietoa syötettäessä. Haastavin osa tätä tehtävää on sen kosketus SAP-ympäristöön. Tämä haaste kuitenkin pystytään ohittamaan seuraavassa osiossa selitetyllä tavalla, mutta pidemmällä aikavälillä olisi hyödyllistä selvittää lupa ohjelmalliseen toimintaan SAP-ympäristössä.

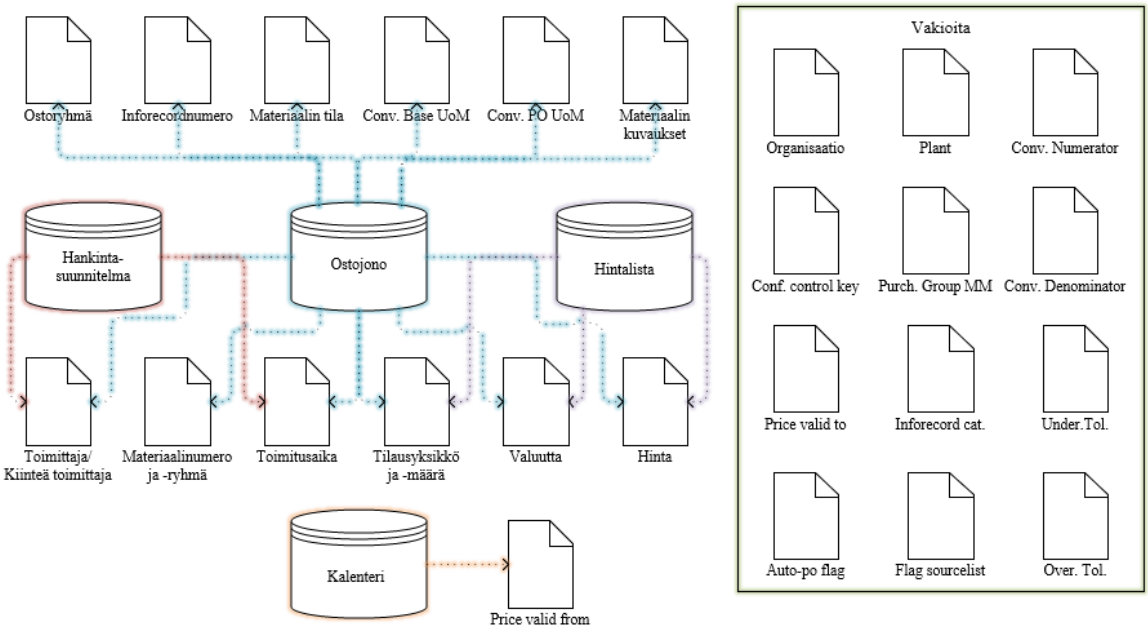
Jotta automaattinen osto menisi läpi, on sillä oltava neljä asiaa tallennettuna SAP:iin: vakiotoimittaja, inforecord, plant-tason inforecord sekä automaatiota varten päivitetty inforecord. Muut kuin kiinteän toimittajan omaavat nimikkeet voidaan suodattaa heti alussa pois, sillä ne vaativat ostajan omaa harkintaa toimittajan valinnassa, jolloin ne eivät sovellu automaatiolle lainkaan. Lopuista nimikkeistä on siis koottava inforecord, plant-tason inforecord sekä automaatiota varten päivitetty inforecord, jos ne puuttuvat. Kuten on useaan kertaan todettu, automaation hyödyntäminen SAP-ympäristössä on tällä hetkellä epäselvää sopimusteknisistä syistä, mutta tiedot voidaan ladata SAP:iin Excel-taulukon (.xlsx) muodossa. Jotta kosketusta SAP-rajapinnan kanssa vältettäisiin, automaatio päätettiin toteuttaa näihin SAP:iin ladattaviin Excel-tiedostoihin. Jotta tiedot latautuisivat SAP:iin oikein, on taulukoiden sarakkeiden oltava tietyssä järjestyksessä ja sisällettävä riittävä määrä tietoa, jotka on eritelty Taulukossa 9 jokaiselle tiedostolle.

Taulukko 9. Tiedot, joita vaaditaan kussakin SAP:iin syötettävässä tiedostossa

Tiedosto/ Sarake	**Inforecord**	**Plant-tason inforecord**	**Päivitetty inforecord**
Materiaalinumero	X	X	X
Materiaalin tila		X	X
Toimittaja	X	X	X
Ostotilauksen määrä		X	X
Tilauksen yksikkö		X	X
Suunniteltu toimitusaika	X	X	X
Suunniteltutoimitusaika MM			X
Pienin ostotilauksen määrä			X
Hinta	X	X	X
Valuutta		X	X

Hintayksikkö	X	X	X
Tilauksen hintayksikkö			X
Organisaatio	X	X	X
Ostoryhmä	X	X	X
Purch. group MM			X
Inforecord cat.			X
Materiaaliryhmä	X	X	X
Materiaaliryhmän kuvaus	X	X	X
Materiaalin kuvaus	X	X	X
Inforecordnumero	X	X	X
Plant	X	X	X
Auto-po flag			X
Price valid from			X
Price valid to			X
Conf. control key			X
Flag sourcelist		X	X
Kiinteä toimittaja	X	X	X
Under.Tol.			X
Over. Tol.			X
Conv. Denominator			X
Conv. PO UoM			X
Conv. Numerator			X
Conv. Base UoM			X

Käyttäjä joutuu edelleen itse manuaalisesti lataamaan tiedostot SAP:iin, mikä ei ole tehokkain ratkaisu, mutta tietojen syöttö kuitenkin nopeutuu, joka taas madaltaa kynnystä nimikkeiden automaattiselle ostolle. Diplomityöhön päätettiin kuitenkin myös tehdä pieni esimerkki siitä, miten ohjelmistorobotiikalla saataisiin helposti automatisoitua ja tehostettua prosessia pidemmälle, jos SAP-rajapinnan automatisaatio olisi sallittua. Demonstraatiota ei testattu varsinaisessa SAP-ympäristössä, vaan siinä käytettiin SAP:in bittikarttoja ja simuloitua ympäristöä, jotka saatiin ruudunkaappauksista. Mallinnusvaiheessa rakennettuun malliin (Liite 8) ja tarkempaan haastatteluun osaprosessin asiantuntijan kanssa perustuen, rakennettiin automatisoitavalle prosessin osalle tarkempi malli tiedostojen vaatimista syötteistä ja niiden lähteistä, jotka on esitetty Kuvassa 13.



Kuva 13. Automaattisen tilauksen tarvitsemat tiedot ja niiden lähteet

Kuten kuvasta nähdään, suurin osa tiedoista saadaan ostojonosta tai sitten ne ovat vakioita. Jos ostojonosta puuttuu tietoa, se voidaan hakea hankintasuunnitelmasta. Hintalista sisältää myös jotain hyödyllisiä tietoja, mutta se tieto on käytännössä aina saatavilla myös ostojonosta, joten hintalistatiedostoista ei tarvitse hakea tietoa tässä ratkaisussa. Sarake ’Price valid from’ on hieman erityinen, sillä se määritellään ohjelman ajon päivämäärän mukaan, joten se vaatii vain pääsyn kalenteritietoihin.

Automaatio toteutettiin Python-ohjelmointikielellä ja erityisesti hyödyntäen OpenPyXL -moduulia, joka on Excel-tiedostojen muokkaamista ja lukua varten luotu kirjasto (Read the Docs 2023a). Myös PyAutoGUI -nimistä moduulia hyödynnettiin ohjelmistorobottidemon luomisessa. Sen työkalukirjasto mahdollistaa muun muassa hiiren ja näppäimistön ohjelmallisen käytön Pythonilla sekä bittikarttojen tunnistuksen graafisen käyttöliittymän kautta (Read the Docs 2023b). Python valittiin ohjelmointikieleksi, koska se on diplomityön tekijälle ennestään tuttu kieli, jolloin automaation työstäminen pystyttiin aloittamaan mahdollisimman nopeasti. Myös VBA-kieltä (Visual Basic Applications) harkittiin, mutta Pythonin todettiin olevan helpompi ja nopeampi toteuttaa sekä parempi kieli tilanteissa, joissa tietoa on haettava Excelin ulkopuolelta. Myös yrityksen olemassa olevat automaatiot ja

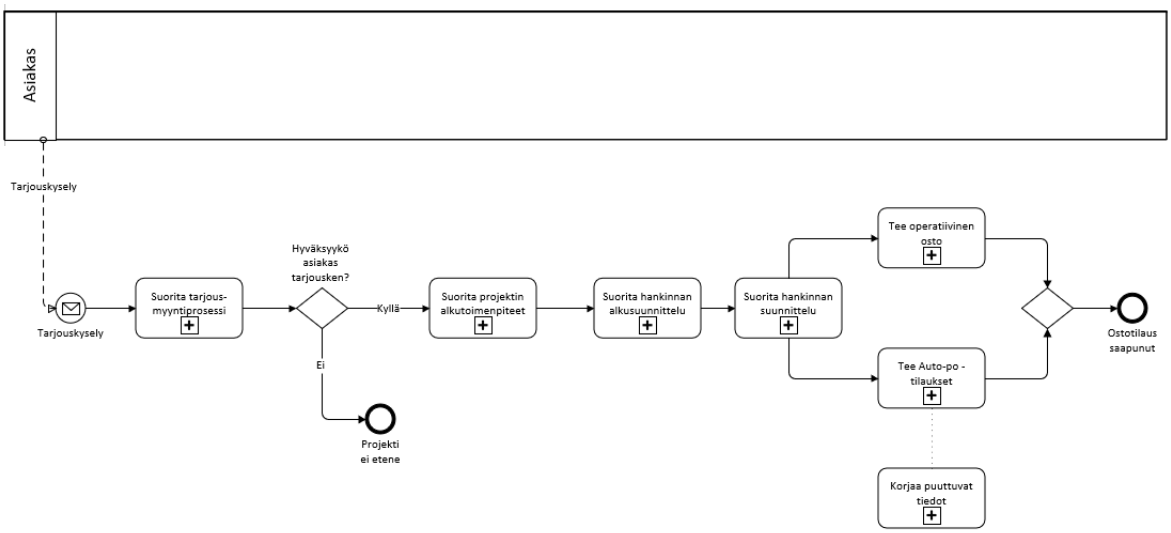
ohjelmistorobotit on ohjelmoitu Pythonilla, joten tämän työn automaatioratkaisu on myös yhdenmukainen niiden kanssa. Automaation rakennuksessa hyödynnettiin IDLE:ä ja Visual Studio Code:a kehitysympäristöinä. Ne ovat ilmaisia ja jo yrityksessä käytössä olevia ohjelmistoja, joten ne täyttävät diplomityölle määritellyt ehdot.

5 Työn ratkaisujen esittely

Tässä luvussa käydään läpi diplomityön aikana syntyneitä tuotoksia, eli pääasiassa tarjoushankintaprosessin mallinnusta ja sen pohjalta tehtyä analyysia sekä automaatiodemon toimintaa. Luvun lopussa myös arvioidaan automaatioratkaisun tuomia suorituskyvyn parannuksia.

5.1 Toteutetun mallinnuksen tulokset

Tutkittava prosessi sijoittuu yrityksen liiketoimintaprosessin alkuun ja on rajattu tarjoushankintaprosessiin. Funktiot, joita tämä prosessi koskettaa ovat myynti, laatu, valmistus, hankinta ja suunnittelu. Tässä välissä oleellisimmat osaprosessit ovat tarjous-myyntiprosessi, projektin alkutoimenpiteet, hankinnan alkusuunnittelu, hankinnan suunnittelu, operatiivinen osto ja automaattiset tilaukset sekä siihen kuuluva puuttuvien tietojen korjaus. Prosessi mallinnettiin ensin karkeasti korkealta tasolta perustuen yrityksen prosessidokumentaatioon sekä aloituspalavereihin. Näiden pohjalta tehtiin kaksi BPMN-mallia, joissa kuvattiin yrityksen tarjous-myyntiprosessi sekä hankintaprosessi. Tämän jälkeen aloitettiin haastattelut pienemmillä porukoilla, joissa jokainen funktion edustaja sai kertoa prosessin kulusta omasta näkökulmastaan, kunnes kaikki osaprosessit saatiin yksitellen mallinnettua tarkasti ja yhdistettyä yhtenäiseen malliin. Korkealta tasolta tarkasteltuna prosessi etenee kuten Kuvassa 14 on esitetty.



Kuva 14. Koko mallinnettu prosessi korkealta tasolta

Prosessi lähtee liikkeelle asiakkaan tarjouskyselystä, joka käsitellään tarjous-myyntiprosessissa. Perustuen tässä osaprosessissa tehtyihin päätöksiin prosessi voi joko päättyä siihen, ettei kauppa etene tai prosessi voi jatkua projektin alkutoimenpiteiden suorittamiseen. Tämän jälkeen projekti jatkuu ensin hankinnan alkusuunnitteluun ja lopulliseen hankinnan suunnitteluun. Hankinnan suunnittelun jälkeen prosessi jakaantuu käsin tehtyyn operatiiviseen ostamiseen sekä automaattisesti tehtäviin tilauksiin. Automaattisen oston osaprosessiin kuuluu myös joskus puuttuvien tietojen korjauksen osaprosessi, josta puhutaan tarkemmin myöhemmin. Kun kaikki tilaukset on tehty, poikkeukset käsitelty ja ostotilaukset saapuneet, prosessi päättyy. Prosessi siis alkaa asiakkaan aloitteesta ja voi päättyä kahdella eri tavalla. Seuraavassa osiossa käydään läpi tarkemmin tärkeimpien osaprosessien sisältöä ja mitä niiden mallinnuksesta opittiin.

Yrityksen tarjous-myyntiprosessi (Liite 2) on suhteellisen vakioitu prosessi. Siitä huolimatta se ei sovellu kovin hyvin automaatiolle, sillä myyjän harkintaa ja asiakkaan kanssa kommunikointia tarvitaan lähes joka käänteessä koko osaprosessin ajan. Automaatiolle on kuitenkin löydetty sijaa hyväksyntäkierrosten käynnistyksessä. Kriittisin tekijä tässä osaprosessissa on asiakkaan vastausnopeus ja sen lähettämän tai pyytämän tiedon täydellisyys. Nämä ovat tekijöitä, joihin yrityksen sisällä ei kovin helposti voida vaikuttaa mutta järjestelmä, jonka kautta rajatun tiedon jakaminen ulkopuolisten toimijoiden, kuten asiakkaiden, kanssa

saattaisi tehostaa osaprosessia. Myös nopea tiedon saatavuus yrityksen sisältä parantaisi suorituskykyä. Tässä osaprosessissa tieto on jaettu useaan eri ohjelmistoon ja järjestelmään eivätkä ne kaikki kommunikoi keskenään kitkattomasti. Projektin alkutoimenpiteet (Liite 3) yhdistävät tarjouksen ja hankinnan suunnittelun tiedottamalla projektin alkamisesta ja varmistamalla tarjouksen ehdot asiakkaan kanssa. Tässä vaiheessa on oleellisinta saada kaikki tarvittava tieto asiakkailta sekä kommunikoida mahdollisesti ilmenevistä muutoksista sisäisesti ja asiakkaalle.

Hankinnan alkusuunnittelussa (Liite 4) tehdään esityö hankinnan suunnittelulle täyttämällä tarvittavat nimikkeiden tiedot. Nimikkeisiin tarvittavaa tietoa haetaan ainakin viidestä eri lähteestä ja mahdolliset korjaukset on pyydettävä myynniltä, joten tiedon kulku ja saatavuus ovat oleellisessa osassa tätä osaprosessia. Koska kyseessä on operaatio, joka koostuu tiedonhausta useasta ohjelmistosta, soveltuisi se teknisesti hyvin automaatiolle ja yhtä sen osa-aluetta harkittiinkin diplomityön automaatiodemoksi. Haasteitakin kuitenkin löytyy, sillä inhimillistä harkintaa tarvitaan halki prosessin ja oleellisin vaikutus osaprosessin suorituskykyyn on asiakkaan jakaman tiedon täydellisyys, johon yrityksen on vaikeaa vaikuttaa. Konfiguraattorin automaattinen tutkinta ja vastaavien nimiketietojen löytäminen sen pohjalta olisi kuitenkin potentiaalinen lisätutkimuksen kohde tämän diplomityön ulkopuolella. Osaluettelon julkaisun jälkeen edetään varsinaiseen hankinnan suunnitteluun (Liite 5). Tässä osaprosessia korostuvat jälleen kerran tiedonhaku useasta erilaisesta lähteestä ja puutteellisten tietojen täyttäminen. Kriittisin vaihe tässä osaprosessissa on varsinaista ajattelutyötä vaativa osien hankinnasta päättäminen ja asiakkaan vaatimien tietojen lähettäminen. Tätä tehtävää voitaisiin tukea automaattisesti esimerkiksi Excel-tiedoston automaattisella täytöllä, joka voitaisiin toteuttaa Excel-macrolla tai Pythonilla. Myös viimeisen vaiheen SAP-merkinnät voitaisiin suhteellisen helposti toteuttaa ohjelmistorobotilla. Nämä vaihtoehdot käsiteltiin aikaisemmin automatisoitavaa prosessin osaa pohdittaessa. Muuten tämän osaprosessin suorituskykyä voitaisiin parantaa paremmalla tietojen ja piirustusten laadunvalvonnalla sekä yhtenäistämällä tarvittavia tiedostoformaatteja.

Hankinnan suunnitelman vapauttaminen käynnistää sekä operatiivisen oston (Liite 6) että automaattisen oston (Liite 7) osaprosessit. Jos keskitytään ensin manuaalisesti tehtäviin

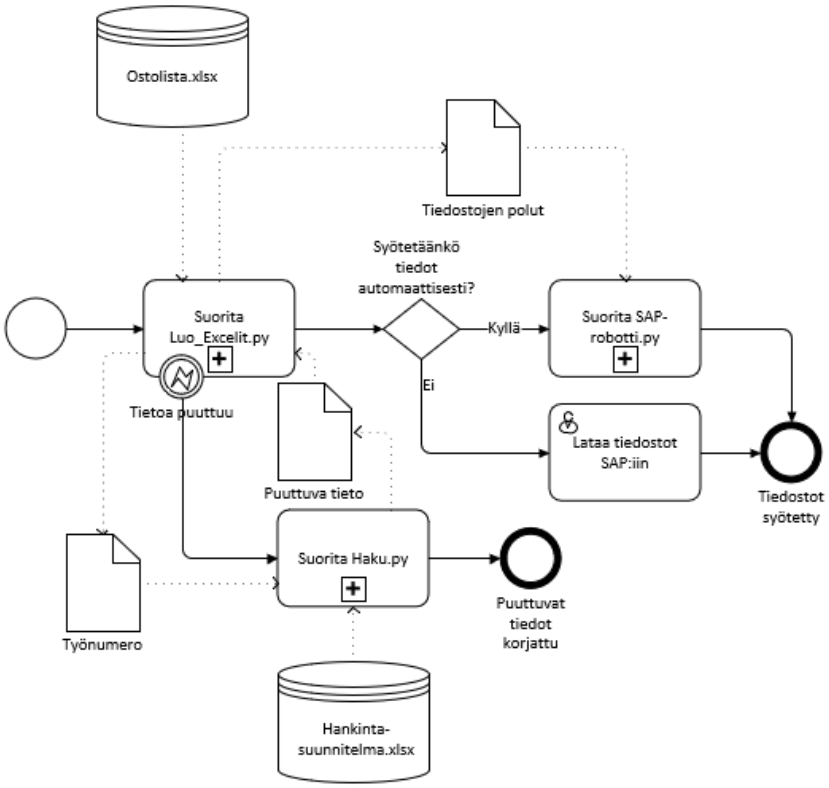
tilauksiin, niissä korostuu erityisesti jatkuva kommunikaatio toimittajan kanssa eri kanavien kautta joko tiedonpyynnön, tiedonjakamisen, erilaisten välitarkastusten tai mahdollisten reklamaatioiden muodossa. Myös jonkin verran tiedonhakua tarvitaan erityisesti osaprosessin alussa, kun ostotilausta ollaan valmistelemassa. Ostajan inhimillistä harkintaa ja kommunikaatiokykyä tarvitaan käytännössä jokaisessa kriittisessä vaiheessa, mutta jokin pieni apuohjelma kuten dokumenttien ja piirustusten automaattinen hakeminen voitaisiin toteuttaa suhteellisen helposti. Paljon suurempi parannus suorituskykyyn saataisiin kuitenkin lisäämällä yrityksessä jo käytössä olevaa automaattisen osto-ohjelman vaikutusaluetta. Yritys tekee jo läheistä yhteistyötä erään perustarvikkeita toimittavan yrityksen kanssa, joka vastaanottaa ostotilaukset automaattisesti. Puutteellisten tai muille toimittajille menevien automaattisten ostotilausten osalta on tällä hetkellä korjattava tarvittavat tiedot käsin (Liite 8), jotta ne lähtisivät automaattisesti matkaan. Kuten aikaisemminkin todettiin, tätä ei ostaja aina ehdi tekemään, joten automaattisen oston potentiaalia jää hyödyntämättä. Tähän ongelmaan haluttiin hakea ratkaisua tässä diplomityössä esitetyssä automaatiodemossa. Sen lisäksi, että demossa vastataan tietyn osaprosessin konkreettiseen ongelmaan, sillä halutaan myös testata yrityksen valmiutta automaatiolle sekä nähdä, miten prosessien mallinnus vaikuttaa sopivan ratkaisun syntyyn.

Mallinnuksessa siis opittiin ja tiivistettiin paljon mielenkiintoista tietoa, jota sitten pystyttiin hyödyntämään suorituskykyä ja laatua parantavilla tavoilla. Tällä tarjous-hankintaprosessin välillä mallinnetuilla osaprosesseilla pystyttiin havainnoimaan joitain yhtenäisyyksiä muun muassa tiedonhaun kriittisyyden ja levinneisyyden sekä tiedonjakamisen ja -vastaanottamisen haasteet ulkopuolisen sidosryhmän kanssa. Tiedon varastointi, ylläpito ja jakaminen ovat korostuneet yrityksessä kuin yrityksessä digitalisaation myötä, mutta niiden hallinta koetaan usein yhä haasteeksi, sillä yrityksiltä usein puuttuu systemaattinen lähestymistapa tiedonhallintaan ja eri ohjelmistojen lisääntyessä, tämä ongelma vain paisuu.

5.2 Automaatiodemon toiminta

Tässä luvussa käydään läpi automaatiodemon toiminnallisuutta tarkemmin. Yksinkertaisuudessaan ratkaisu luo, täyttää ja muotoilee kolme Excel-tiedostoa sellaisiksi, että käyttäjä voi vain ladata ne SAP:in transaktioihin siltään, jonka jälkeen ostotilaukset lähtevät

automaattisesti. Joidenkin sarakkeiden tiedot ovat vakioita ja toisten tietoa on haettava ulkoisesta lähteestä, kuten ostojonosta tai hankintasuunnitelmasta. Tässä luvussa on myös demonstraatio tiedostot automaattisesti SAP:iin lataavasta ohjelmistorobotista. Automaatiodemo siis koostuu kolmesta Python-ohjelmasta (Luo_Excelit.py, Haku.py ja SAP_robotti.py) ja kahdesta tietolähteestä (Ostolista.xlsx ja Hankintasuunnitelma.xlsx), joiden keskinen toiminta on esitetty Kuvassa 15.



Kuva 15. Automaatiodemon ohjelmien ja tiedonkulun mallinnus

Vähintään ohjelma Luo_Excelit.py suoritetaan, mutta jos siitä puuttuu tiettyjä hankintasuunnitelmasta löytyviä tietoja, ne haetaan suorittamalla ohjelma Haku.py. Puuttuvat tiedot haetaan hankintasuunnitelmasta työnumeroiden perusteella. Jos tiedostot halutaan liittää automaattisesti, voitaisiin lopuksi vielä suorittaa ohjelma SAP-robotti.py, joka saisi tiedostojen polut ohjelmasta Luo_Excelit.py. Tämä päätös on esillä kuitenkin vain demonstraatiotarkoituksena, koska oikeaa SAP-ympäristöä ei voida tällä hetkellä automatisoida, joten käyttäjän on itse ladattava luodut tiedostot SAP:iin. Käyttäjän on myös ladattava omalle koneelleen

.xlsx-versiot ostojonotiedostosta ja uusin versio hankintasuunnitelmasta, sillä ohjelmat käyttävät niitä tietolähteinään. Seuraavaksi tarkastellaan tarkemmin itse ohjelmakoodin toimintaa.

5.2.1 Luo Excel-tiedostot

Liitteessä 9 on kuvattu osa Luo_Excelit.py lähdekoodin alkua, jossa kutsutaan ohjelmassa tarvittavat moduulit ja määritellään ostojonotiedostoon johtava polku. Kaikki koodissa käytetyt tiedostopolut sekä muu luottamuksellinen tieto on peitetty tietoturvasyistä. Tässä ohjelmassa hyödynnetään runsaasti OpenPyXL-moduulia, sillä se sisältää laajan kirjaston erilaisia funktioita ja komentoja Excel-tiedostojen muokkaamiseen. Funktion load_workbook avulla voidaan määritellä Excel-työkirja (wb) ja sen sivu (sheet). Sivuksi määritellään työkirjan aktiivinen sivu, joka tässä työkirjassa tarkoittaa vain sen ensimmäistä ja ainoaa sivua. Ostojonotiedostoa käytetään pohjana luotaville taulukoille, sillä se sisältää jo suuren osan niiden tarvitsemasta tiedosta. Lisäksi lähdekoodin suorittamiseen kuluva aika mitataan asettamalla alkamisajankohta muuttujaan alku ja vähentämällä se myöhemmin ohjelman lopussa asetettavasta lopetusajasta. Tästä syystä myös time-moduulia on kutsuttava ohjelman alussa.

Aluksi määritellään myös kaikki ohjelman myöhemmin tarvitsemat aliohjelmat, jotka näkyvät myös Liitteessä 9. Yleisesti aliohjelmat koskevat tiedonhakua, vakiotiedon täyttämistä kaikille sarakkeen riveille, uusien sarakkeiden lisäämistä sekä sarakkeiden kopiointia ja liittämistä työkirjan sisällä. Seuraavaksi käsitellään tiedonhakufunktion tarkempaa toimintaa. Aliohjelmaan hae_tieto syötetään neljä parametria, jotka ovat työkirjan sivu ja sarake, joista tieto haetaan, sekä työkirjan sivu ja sarake, joihin tiedot kirjoitetaan. Näiden perusteella iteroidaan annetun työkirjan sivun kaikki sarakenimet läpi etsien luettavan sarakkeen nimeä ja sen löydyttyä, kaikki sarakkeen solujen arvot paitsi sarakkeen nimi lisätään arvot nimiseen listamuuttujaan, joka luotiin tyhjänä funktion alussa. Seuraava iteraatio alkaa samalla tavalla, mutta tällä kertaa iteroidaan kirjoitettavan sivun sarakenimet läpi etsien kirjoitettavan sarakkeen nimeä. Kun kirjoitettava sarake löytyy, kaikki arvot-listan arvot kirjoitetaan sarakkeen soluihin rivi kerrallaan aloittaen toisesta rivistä.

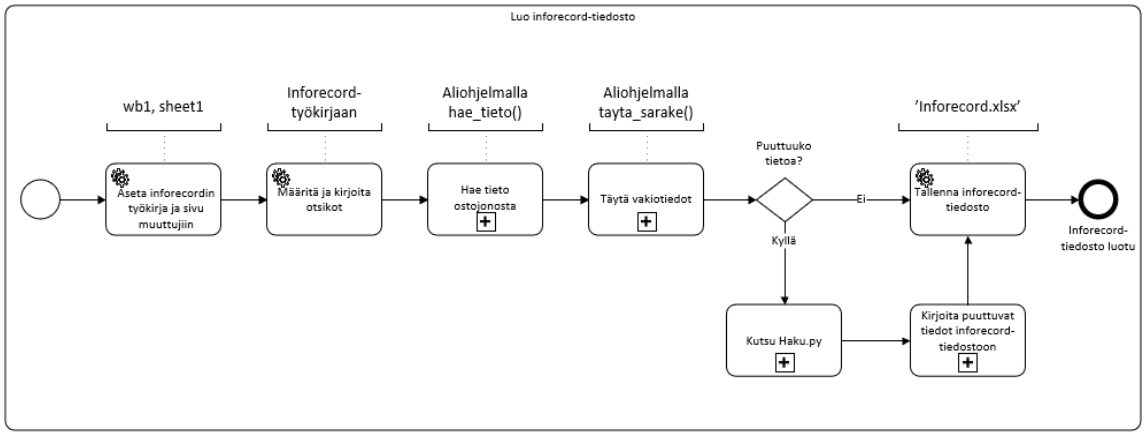
Seuraavaksi määritellään aliohjelma vakiotietojen täyttämiseksi sarakkeisiin. Funktio vaatii kolme parametria, jotka ovat työkirjan sivu, sarakkeen nimi ja kirjoitettava vakioarvo. Näiden perusteella aliohjelma iteroi kaikki annetun sivun sarakenimet läpi, kunnes annettu sarakkeen nimi löytyy, jolloin se täyttää kaikki sarakkeen solut annetulla arvolla. Arvoja kirjoittaessaan toistorakenne välttää sarakeotsikon päälle kirjoittamisen tarkastamalla, onko iteraation indeksi nolla ja ohittamalla kirjoitusvaiheen, jos se on. Ennen tätä kirjoittamista funktio ottaa huomioon, onko solu tyhjä tai jos solussa lukee ’FIM’ tai ’FIL’. Täytettyihin soluihin ei tarvitse kirjoittaa vakioarvoa uudelleen paitsi arvojen ’FIM’ ja ’FIL’ tapauksessa, sillä ne on korvattava toisella vakioarvolla. Sarakkeen lisäyksen aliohjelmaan syötetään parametrit työkirjan sivulle, sarakkeen indeksille ja sarakkeen nimelle. Funktio yksinkertaisesti luo uuden sarakkeen annetulle työkirjan sivulle annetun indeksin kohdalle ja kirjoittaa annetun sarakkeen nimen sarakkeen ensimmäiselle riville. Sarakkeita ei poistu työkirjasta, vaan indeksin oikealle jäävät sarakkeet vain siirtyvät eteenpäin yhdellä indeksillä.

Sarakkeita kopioivaan aliohjelmaan syötetään kolme parametria, jotka ovat kopioitavan sarakkeen nimi, kirjoitettavan sarakkeen nimi ja se työkirjan sivu, josta nämä sarakkeet löytyvät. Funktio alkaa annetun sivun sarakenimien iteraatiolla, kunnes kirjoitettavan sarakkeen nimi löytyy, jolloin sarakkeen indeksi otetaan talteen. Tämän jälkeen sivun sarakenimet iteroidaan jälleen, kunnes kopioitavan sarakkeen nimi löytyy. Lopuksi voidaan vain rivi kerrallaan kopioida tästä sarakkeesta löytyvät arvot aikaisemmin löydettyyn sarakeindeksiin, ohittaen kuitenkin sarakkeen nimen kopioinnin ja sen päälle kirjoittamisen.

Seuraavaksi ostojono siivotaan turhista riveistä. Koska nyt ollaan kiinnostuneita vain nimikkeistä, joilla on vakiotoimittaja, kaikki rivit, joilla ei ole merkittynä vakiotoimittajaa poistetaan. Tämä tehdään ensin luomalla tyhjä listamuuttuja poistettavista riveistä (poista_rivit), jonka jälkeen kaikki Fixed Vendor -sarakkeen rivit iteroidaan läpi. Jos arvo on tyhjä, rivin indeksi tallennetaan poistettavien rivien listaan ja jos riviltä löytyy arvo, jatkuu iteraatio seuraavaan riviin. Kun kaikki rivit on käyty läpi, kaikki poistettavien rivien listalle tallennetut rivit poistetaan sivulta rivi-indeksin mukaan käänteisessä järjestyksessä funktiolla delete_rows. Järjestyksen on oltava käänteinen, sillä muuten indeksit muuttuvat ja jokainen poistettu rivi sekoittaa seuraavan iteraation, esimerkiksi jos toinen rivi poistetaan,

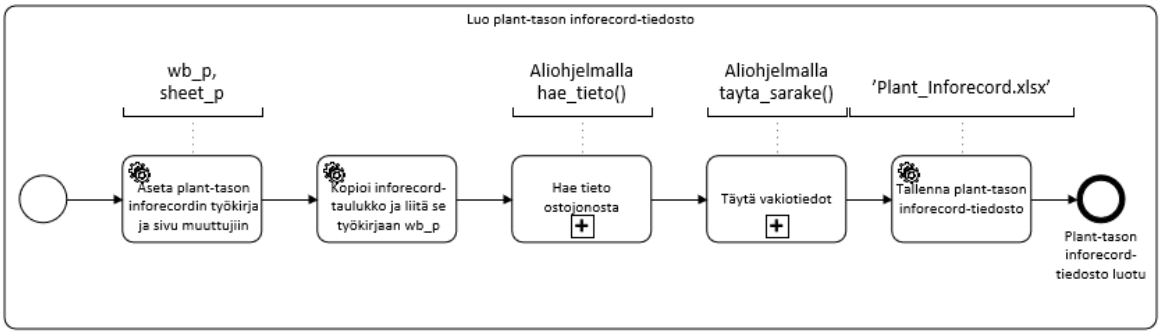
kolmannesta rivistä tulee toinen rivi ja neljännestä kolmas jne. Turhien rivien poistamisen jälkeen otetaan talteen työnumerot omalle listalleen (tyonro), sillä niitä saatetaan tarvita myöhemmin.

Seuraavaksi luodaan ensimmäinen kolmesta Excel-tiedostosta aloittamalla uuden työkirjan (wb1) ja sivun (sheet1) asettamisella. Inforecord-tiedoston sarakeotsikot on määritelty halutussa järjestyksessä otsikot-listaan. Jos otsikoita halutaan lisätä tai muuten muuttaa tässä tiedostossa, se on tehtävä tämän listamuuttujan kautta. Sarakkeiden otsikot kirjoitetaan iteroiden otsikot-lista läpi inforecord-työkirjan ensimmäiselle riville. Nyt kun pohja inforecord-tiedostolle on luotu, voidaan sen täyttäminen aloittaa. Suurin osa inforecordin tiedoista saadaan suoraan ostojonotiedostosta aliohjelmalla hae_tieto. Seuraavaksi täytetään mahdollisesti puuttuvat vakiotiedot aliohjelman tayta_sarake avulla. Tämän jälkeen tarkastetaan, onko tiettyjen sarakkeiden arvoissa puutteita ja jos on, haetaan tämä puuttuva tieto hankintasuunnitelmasta kutsumalla hakuohjelmaa ja kirjoittamalla puuttuvat arvot inforecord-tiedostoon. Kutsuttu hakuohjelma esitellään tarkemmin myöhemmin tässä luvussa. Tässä kohdassa suurin osa sarakkeista jää tyhjiksi, mutta niiden on oltava olemassa, jotta tiedot latautuvat oikein SAP:iin. Lopuksi työkirja tallennetaan funktiolla save ja nimellä ’Inforecord.xlsx’ samaan kansioon, josta ohjelma löytyy. Koska taulukko tallennetaan eri nimellä ja eri paikkaan, ei lähdetiedostona toimineeseen ostojonotiedostoon tapahdu minkäänlaisia muutoksia. Tämä on yleisesti hyvä käytäntö ainakin suuria muutoksia tehdessä, sillä virheellinen ohjelman ajo saattaa tehdä lähdetaulukosta lukukelvottoman tai pahimmassa tapauksessa ajo voi jopa korruptoida sen. Inforecord-tiedoston luominen, muotoilu ja täyttäminen on tiivistetty Kuvaan 16, jossa se on esitetty BPMN-syntaksilla avattuna osaprosessina.



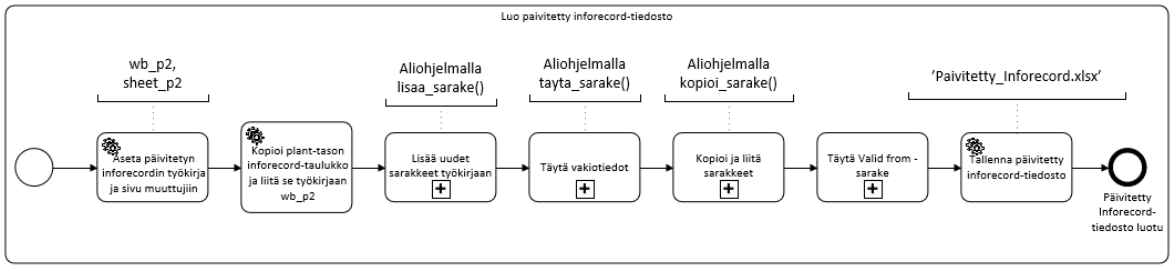
Kuva 16. Inforecord-tiedoston luominen

Seuraavan Excel-taulukon, eli plant-tason inforecordin, luonti alkaa jälleen uuden työkirjan (wb_p) ja sivun (sheet_p) asettamisella. Tämän taulukon pohjana käytetään aikaisemmassa vaiheessa luotua inforecord-taulukkoa, joten se kopioidaan ja liitetään uuteen työkirjaan. Rivien ja sarakkeiden rajat määritellään aikaisemman taulukon koon mukaan, jonka jälkeen kaikki rivit ja sarakkeet kopioidaan toistorakenteen avulla uuteen plant-tason inforecord-taulukkoon. Seuraavaksi täytetään aliohjelmilla tarvittavat tiedot ostojonosta ja mahdollisesti puuttuvat vakiotiedot kuten aikaisemman taulukon kanssa. Lopuksi työkirja tallennetaan nimellä 'Plant_Inforecord.xlsx'. Nyt ohjelmistokansiossa on kaksi Excel-tiedostoa ja ostojonotiedosto on edelleen koskemattomana lähtökansiossaan. Plant-tason inforecord-tiedoston luominen, muotoilu ja täyttäminen on tiivistetty Kuvaan 17, jossa se on esitetty BPMN-syntaksilla avattuna osaprosessina.

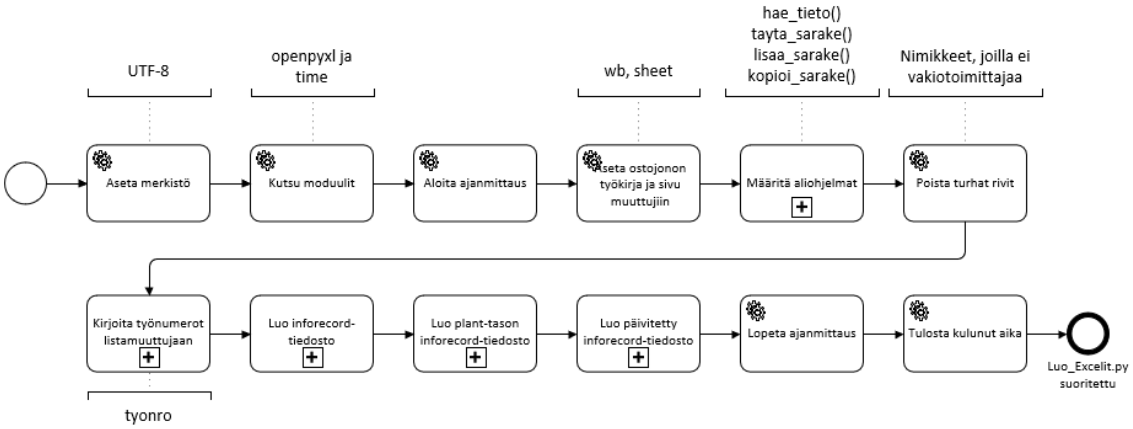


Kuva 17. Plant-tason inforecord-tiedoston luominen

Viimeinen Excel-tiedosto, eli päivitetty inforecord, luodaan seuraavaksi. Samalla tavalla kuin aikaisemmat tiedostot, tämä osio alkaa uuden työkirjan (wb_p2) ja sen sivun (sheet_p2) asettamisella sekä aikaisemmassa kohdassa luodun taulukon kopioinnilla tähän uuteen työkirjaan. Kun uudet sarakkeet on lisätty oikeille paikoilleen, täytetään tarvittavat sarakkeet, joko vakioarvoilla tai muista sarakkeista kopioiduilla arvoilla hyödyntäen alussa määriteltyjä aliohjelmia lisaa_sarake, tayta_sarake ja kopioi_sarake. Valid from -sarake vaatii joitain erikoistoimenpiteitä, jotta data menisi SAP:ista oikein läpi. Saraketta varten on määriteltävä ohjelman ajopäivän päivämäärän muodossa YYYYMMDD. Tätä varten kutsutaan datetime-moduuli, jotta ’tänään’ saadaan määriteltyä muodossa YYYY-MM-DD. Tämä päivämäärä muutetaan tekstimuotoon ja muokataan haluttuun muotoon poistamalla väliviivat. Lopuksi muokatut tiedot kirjoitetaan kaikkiin Valid from -sarakkeen soluihin. Viimeinen tiedosto tallennetaan nimellä ’Paivitetty_Inforecord.xlsx’ lähdekoodin kanssa samaan kansioon ja nyt kaikki tarvittavat Excel-tiedostot ovat valmiina ladattavaksi SAP:iin. Päivitetyn inforecord-tiedoston luominen, muotoilu ja täyttäminen on tiivistetty Kuvaan 18, jossa se on esitetty BPMN-syntaksilla avattuna osaprosessina. Tämän jälkeen esitetään Kuva 19, johon on tiivistetty koko edellä kuvatun ohjelman suoritus.



Kuva 18. Päivitetyn inforecord-tiedoston luominen



Kuva 19. Luo_Excelit.py tiivistettynä

Tiedostojen luonnit on Kuvassa 19 esitetty suljettuina osaprosesseina. Ohjelman lopuksi tulostetaan vielä suorituksessa kulunut aika, jota tarkastellaan myöhemmin tässä luvussa.

5.2.2 Hae tietoa hankintasuunnitelmasta

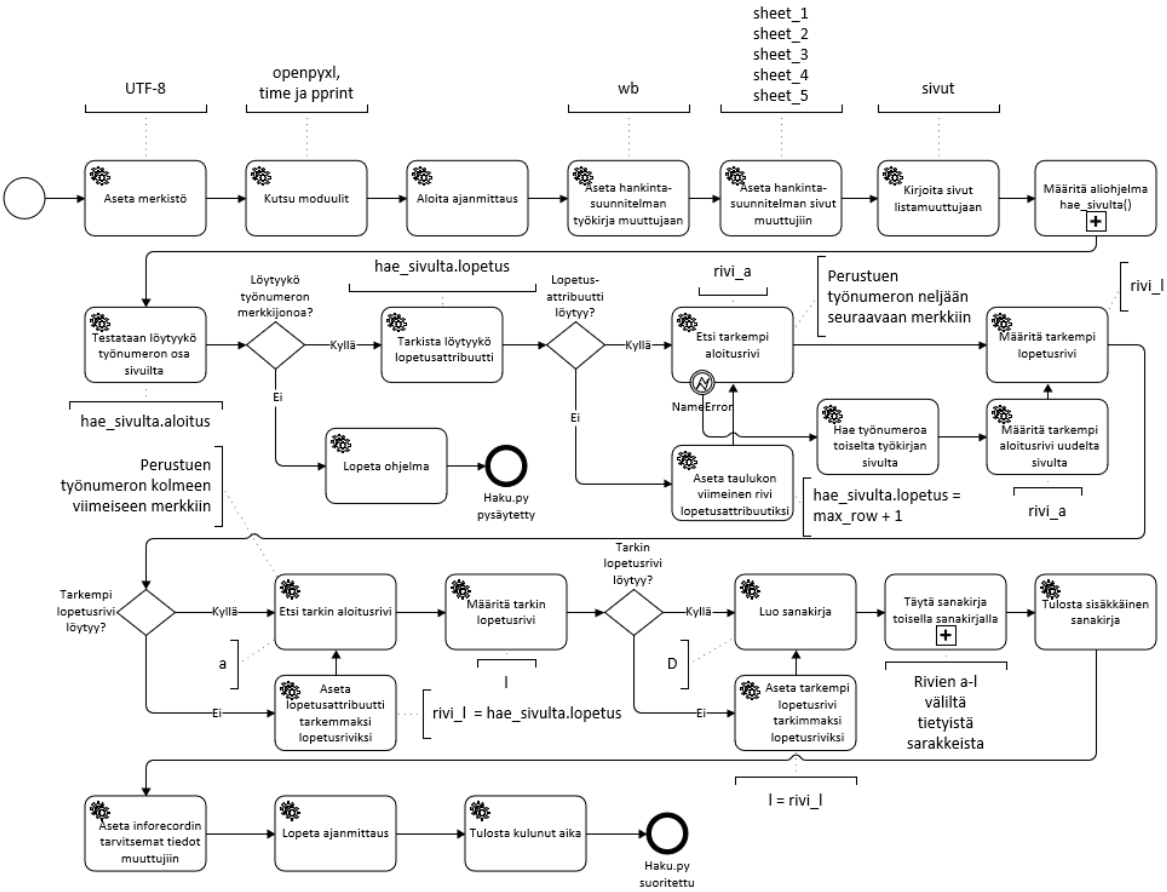
Tässä luvussa esitellään tarkemmin Excel-tiedostojen luonnissa mahdollisesti kutsuttua hakuohjelmaa Haku.py. Liitteessä 10 on kuvattu ohjelman alkutoimenpiteet sekä sen aliohjelman määrittely. Tiivistetysti ohjelma hakee tietoa hankintasuunnitelmasta ostojonosta saatavan työnumeron perusteella. Työnumerot ovat aina samassa muodossa ja sisältävät aina saman määrän merkkejä esimerkiksi P-AA-AAAAAA-AAA-BBB1-CC3. Kuvassa 20 on esitelty hankintasuunnitelman rakennetta ja sitä, missä muodossa työnumerot löytyvät taulukosta. Kuvan hankintasuunnitelmaa ja työnumeroiden merkkejä on muokattu sekä projektien, asiakkaiden, materiaalien ja osien nimet on piilotettu niiden sisältämän luottamuksellisen tiedon vuoksi. Ensin on tarkasteltava työnumeron viittätoista ensimmäistä merkkiä (esim. P-AA-AAAAAA-AAA), jotta löydetään oikea sivu ja oikea kohta taulukosta, joka Kuvassa 20 on rivien 494 ja 520 välillä. Sitten on tarkasteltava neljää seuraava merkkiä (esim. BBB1), jotta löydetään tarkempi tarkasteluväli, joka Kuvassa 20 löytyy rivien 495 ja 507 väliltä. Lopuksi tarkennetaan vielä kolmen viimeisen merkin perusteella (esim. CC3), joka Kuvassa 20 tarkoittaa väliä 502–503. Ohjelmallisesti rajauksen viimeistä riviä ei tarkastella, vaan se toimii rajana, joten käytännössä tässä tarkoitetaan vain riviä 502. Ohjelma tarkentaa vastaavasti tarkasteluväliä, kunnes löydetään haluttu työnumero, jonka jälkeen

työnumeron arvot kirjoitetaan sisäkkäiseen sanakirjaan, josta tiedot voidaan helposti hakea tarvittaessa.

494	PROJEKTI	P-AA-AAAAAA-AAA		
495	ASIAKAS	BBB1		
496		Part 1	CC1	Material a
497		Part 2	CC2	Material a
498				Material b
499				Material c
500				Material d
501				Material e
502		Part 3	CC3	Material a
503		Part 4	CC4	Material a
504		Part 5	CC5	Material a
505		Part 6	CC6	Material a
506		Part 7	CC7	Material a
507	ASIAKAS	BBB2		
508		Part 1	CC1	Material a
509		Part 2	CC2	Material a
510				Material b
511				Material c
512				Material d
513				Material e
514		Part 3	CC3	Material a
515		Part 4	CC4	Material a
516		Part 5	CC5	Material a
517		Part 6	CC6	Material a
518		Part 7	CC7	Material a
519				
520	PROJEKTI	P-BB-BBBBBB-BBB		
521	ASIAKAS	BBB1		

Kuva 20. Hankintasuunnitelman rakenne

Ohjelma Haku.py (Kuva 21 ja Liite 10) alkaa wbs-muuttujan määrittelyllä ja tarvittavien apumoduulien kutsumisella. Muuttujan wbs arvo on työnumero, joka saadaan edellisen ohjelman (Luo_Excelit.py) tyonro-listalta. OpenPyXL on keskeinen moduuli myös tässä ohjelmassa, sillä hankintasuunnitelma on Excel-tiedosto. Lisäksi kutsutaan time-moduulia ajan mittaamiseen ja pprint-moduulia sanakirjojen (dictionary) selkeämpään esittämiseen. Seuraavaksi määritellään työkirjaksi hankintasuunnitelma liittämällä sen sijainnin polku ja parametri ’Data_only=True’. Tämä ominaisuus varmistaa, että solun arvo luetaan itsessään sen kaavan sijaan. Hankintasuunnitelma sisältää useita sivuja, joista voi löytyä haluttua tietoa, joten sivut asetetaan muuttujiksi ja lisätään listamuuttujaan sivut.



Kuva 21. Haku.py tiivistettynä

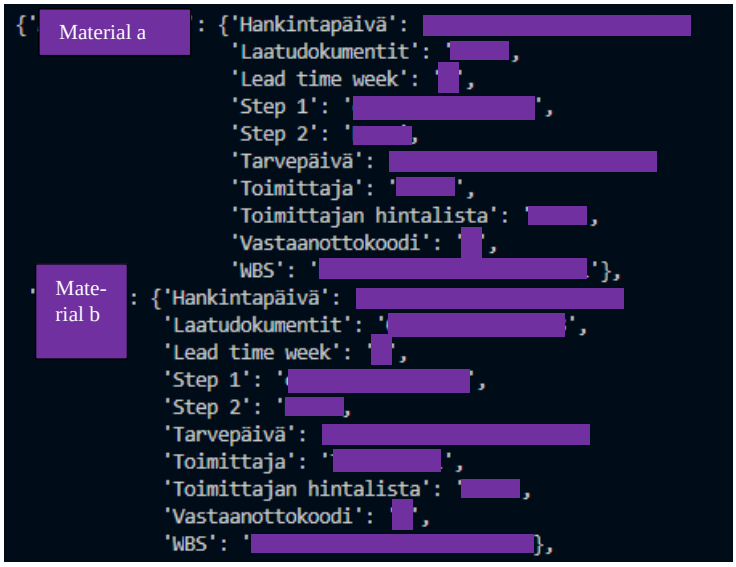
Seuraavaksi määritellään aliohjelma, joka hakee annetulta sivulta työnumeron viiteentoista ensimmäiseen merkkiin perustuvan merkkijonon ja merkitsee sen aliohjelman aloitusattribuutiksi. Tähän aloitusattribuuttiin perustuen haetaan lopetusattribuutti sen perusteella, milloin löydetään seuraava työnumero aloitusrivin jälkeen. Esimerkiksi Kuvassa 20, jos työnumeron viisitoista ensimmäistä merkkiä ovat P-AA-AAAAAA-AAA, tarkasteltava väli tarkentuisi rivien 494-520 välille aliohjelman ajon jälkeen. Seuraavassa vaiheessa etsitään työnumeron viittätoista ensimmäistä merkkiä kaikilta työkirjan sivuilta hyödyntäen äsken määriteltyä aliohjelmaa. Aliohjelman ajon jälkeen testataan, löytyikö aloitusattribuutti tarkasteltavalta sivulta ja jos sitä ei löydy, syntyy virhe (AttributeError), joka laukaisee sivut-listan seuraavan iteraation. Jos sivu taas löytyy, se tallennetaan muuttujaan s ja iteraatio loppuu. Lopuksi tulostetaan käyttäjälle, löytyikö työnumeron osaa sivuilta ja jos sitä ei löytynyt, ohjelma pysäytetään hallitusti. Seuraavaksi testataan lopetusattribuutin olemassaoloa ja jos sitä ei löydy, sivun viimeistä riviä käytetään lopetusattribuutin määrittelemiseksi. Syy siihen,

miksi lopetusattribuuttia ei pystytä aina löytämään on se, että etsittävä työnumero on sivun viimeinen, jolloin lopetusattribuuttia ei voida määritellä seuraavan työnumeron perusteella.

Seuraavaksi tarkennetaan tarkasteltavaa väliä perustuen työnumeron neljään seuraavaan merkkiin. Uusi tarkempi aloitusrivi (rivi_a) etsitään aikaisemmin määritellyltä väliltä. Tarkempi lopetusrivi (rivi_l) löytyy etsimällä seuraavaa merkkijonoa iteroimalla rivi_a:n ja lopetusattribuutin väliltä. Esimerkiksi Kuvassa 20, jos työnumeron neljä seuraavaa merkkiä ovat BBB1, tarkentuisi tarkasteluväli riveille 495-507. Jos muuttujan rivi_a käsittely laukaisee virheen (NameError), työnumeron ensimmäistä väliä haetaan toiselta sivulta ja rivi_a pyritään määrittelemään uudelleen. Tämä käsittely on tehtävä siinä suhteellisen harvinaisessa tapauksessa, jossa työnumeron viisitoista ensimmäistä merkkiä löytyvät useammalta työkirjan sivulta ja siten työnumeron seuraavat neljä merkkiä eivät löydykään ensin löytyvältä sivulta. Kun uusi sivu ja tarkasteluväli on määritelty, se voidaan tarkentaa määrittelemällä ensin rivi_a ja sitten rivi_l samalla tavalla kuin aikaisemmin. Jälleen kerran, jos kyseessä on viimeinen tarkempi merkkirivi, rivi_l ei ole määriteltävissä, jolloin aliohjelman hae_sivulta lopetusattribuutista tehdään tarkempi lopetusrivi. Esimerkiksi Kuvassa 20, jos työnumeron neljä seuraavaa merkkiä olisivat BBB2, niin tarkasteltavalla välillä rivi_a olisi 507, lopetusattribuutti olisi 518 ja rivi_l olisi määrittelemätön, jolloin lopetusattribuutista tulisi rivi_l, koska BBB2 on viimeinen merkkijono tarkasteltavassa sarakkeessa tarkasteltavan työnumeron alla.

Tarkasteltavaa väliä tarkennetaan vielä kerran perustuen työnumeron kolmeen viimeiseen merkkiin. Tarkasteluvälinä käytetään edellisessä kohdassa saatuja arvoja rivi_a ja rivi_l ja kun kolme viimeistä merkkiä löytyvät tältä väliltä, se tallennetaan muuttujaan a. Esimerkiksi Kuvassa 20, jos työnumeron kolme viimeistä numeroa olisivat CC2, niin a:n arvoksi saataisiin rivi 497. Tämän perusteella voidaan etsiä välin loppu, joka merkitään muuttujaan l. Kuvassa 20 kolmen viimeisen numeron ollessa CC2 muuttujan l arvo olisi rivi 502. Tämän jälkeen tarkastetaan, löytyikö muuttujaa l ja jos ei löytynyt, niin muuttujan rivi_l arvo asetetaan sen arvoksi. Tässä on sama logiikka kuin kahdessa aikaisemmassa loppuarvon tarkastuksessa ja korjauksessa.

Nyt kun tarkka työnumeron väli (a-l) on löydetty, voidaan näiltä riveiltä kirjata ylös kaikki haluttu tieto. Tiedot kirjoitetaan sisäkkäiseen sanakirjaan D, joka luodaan tyhjänä tämän osion alussa. Poikkeuksena ovat ne nimikkeet, joita ei olla merkitty ostettaviksi, joten ne ohitetaan. Ulomman sanakirjan avaimeksi määritellään työvaihetta tai hankittavaa osaa kuvaavan sarakkeen arvo, sillä ne ovat uniikkeja jokaisella tarkalla työnumerovälillä. Sisemmän sanakirjan avaimiksi asetetaan tarkasteltavien sarakkeiden nimet. Arvot kirjoitetaan sanakirjaan rivi kerrallaan näistä hankintasuunnitelman sarakkeista, kunnes päästään tarkasteluvälin loppuun. Luotu sisäkkäinen sanakirja voidaan tulostaa tässä kohdassa tarkastusta varten Kuvan 22 osoittamassa muodossa.



Kuva 22. Täytetty sisäkkäinen sanakirja tulostettuna (luottamuksellinen tieto peitetty)

Lopuksi asetetaan vielä Excelien kokoamiseen mahdollisesti tarvittavat tiedot omiin muuttujiinsa, joita aikaisempi ohjelma voi tarvittaessa kutsua. Myös muutama muu tieto tallennetaan muuttujiin mahdollista jatkokehitystä varten. Aivan ohjelman lopuksi mitataan vielä suoritukseen mennyt aika, jota käsitellään myöhemmin tässä luvussa.

5.2.3 Ohjelmistorobotti SAP-syöttöihin

Diplomityössä haluttiin vielä demonstroida ohjelmistorobottien tehokkuutta tiedonsyötössä, mutta koska lupa SAP-ympäristössä toimimiseen robottien kanssa on tällä hetkellä epäselvää, rakennettiin työhön Excelissä simuloitu SAP-ympäristö. Koska ohjelmistorobotit toimivat graafisen käyttöliittymän kautta, on simuloituun ympäristöön tehty robotti toiminnaltaan käytännössä sama kuin oikeaan ympäristöön rakennettu robotti, kunhan simuloitu ympäristö vain näyttää samalta ja toimii samalla logiikalla kuin oikea ympäristö. Simulaatio tehtiin Exceliin perustuen kuvankaappauksiin varsinaisesta SAP-ympäristöstä ja lisäämällä toimivia tekstikenttiä, nappeja ja valintaruutuja niiden oikeille paikoille. Liitteissä 12, 13 ja 14 on esitetty kaikki sivut, joissa ohjelmarobotti toimii. Seuraavaksi käydään tarkemmin läpi itse ohjelmakoodia.

Ohjelma SAP_robotti.py alkaa tarvittavien moduulien kutsumisella ja ajanmittauksen aloittamisella. Liitteessä 11 on esitetty osa ohjelmistorobotin lähdekoodista, kuten sen alkutoimenpiteitä ja hiiren liikuttamisen logiikkaa perustuen suhteellisiin välimatkoihin. PyAutoGUI-moduuli sisältää käytännössä kaikki työkalut, joita tässä ohjelmassa käytetään ja time-moduuli kutsutaan ajanmittausta ja taukoja varten. PyAutoGUI-moduuli tarjoaa kirjaston työkaluja muun muassa bittikarttojen tunnistukseen, hiiren ohjelmalliseen liikuttamiseen ja näppäimistön ohjelmalliseen käyttöön. Aluksi varmistetaan myös, että robotin pysäyttämisen turvamekanismi on päällä ja ilmoitetaan käyttäjälle, miten pysäytys tehdään. Seuraavaksi käyttäjää pyydetään kertomaan, kuinka monta tiedostoa ollaan syöttämässä ja tämä syöte tallennetaan muuttujaan t. Jos muuttujan t arvo on 0, ohjelma pysäytetään. Jos muuttujan t arvo on jotain muuta kuin 0, 1, 2 tai 3, käyttäjää pyydetään syöttämään uusi arvo. Kun muuttujan t arvoksi saadaan hyväksyttävä arvo, ohjelma jatkaa seuraavaan vaiheeseen.

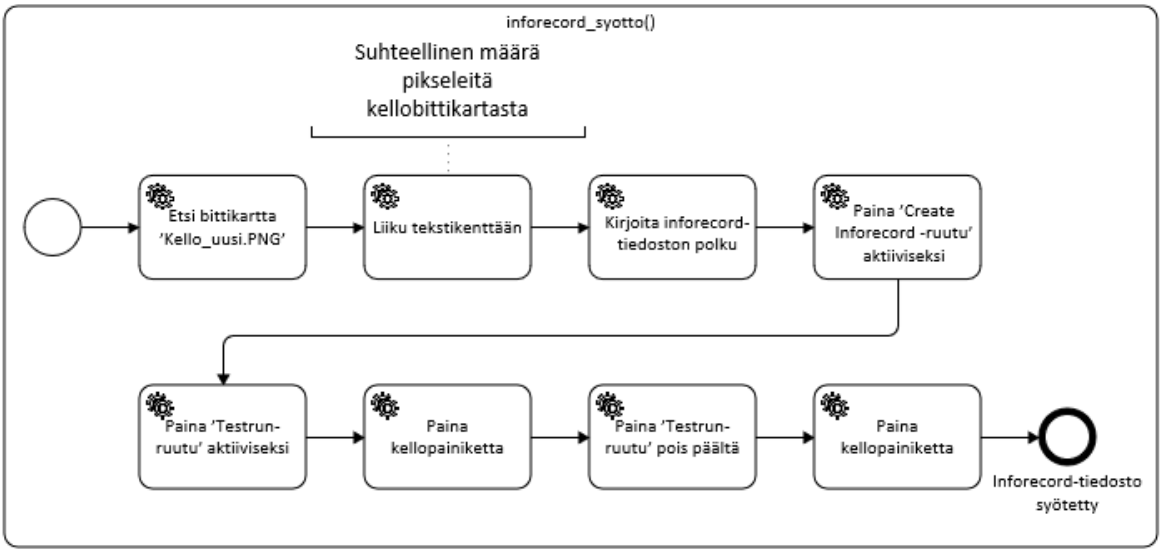
Ohjelman käynnistyessä SAP:in ’Production System -etusivun’ on oltava auki ja kokonaan näkyvillä tietokoneen ruudulla (Liite 12). Tietoturvan takia robotti ei kirjaudu SAP:iin käyttäjän tunnuksilla, vaan käyttäjän on itse kirjauduttava sisään ennen ohjelman käynnistystä. Ohjelma alkaa bittikartan ’Check.PNG’ etsimisellä ruudulta ja sen löytyessä hiiri siirretään sen kohdalle. Kaikki ohjelman käyttämät bittikartat nimineen on esitetty Taulukossa 10.

Taulukko 10. SAP-robotissa käytettyjen bittikarttojen nimet ja kuvat

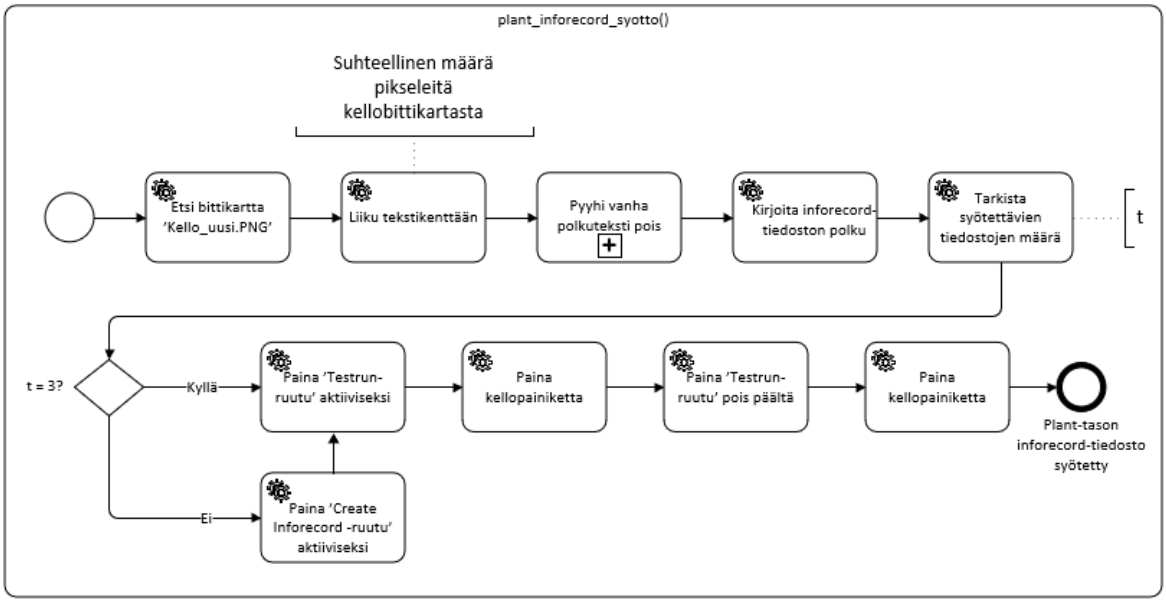
Nimi	Bittikartta
Check.PNG	
Kello_uusi.PNG	
Takaisin.PNG	

Kun hiiri on siirtynyt bittikartalle, se siirretään suhteessa tästä kohdasta 50 pikseliä oikealle, jonka jälkeen ohjelma painaa hiiren vasenta painiketta kahdesti ja kirjoittaa tekstin ’ZMMINFOREC’. Tässä kohdassa on hyvä huomauttaa, että nämä suhteelliset hiiren liikkeet ovat riippuvaisia näytön resoluutiosta eivätkä ne toimi oikein, jos robottia yritetään ajaa eri resoluutiolla kuin missä se on alun perin määritelty. Tämän diplomityön robotti tehtiin näytön resoluutiolle 3440 x 1440 pikseliä ja jos sitä halutaan testata muun kokoisilla näytöillä, on hiiren liikkeet ensin suhteutettava uudelle resoluutiolle. Seuraavaksi ohjelma siirtää hiiren takaisin bittikartan kohdalle ja klikkaa, mikä vaihtaa sivun ’ZMMINFOREC-transaktioon’ (Liite 13). Tämän jälkeen ohjelma odottaa puoli sekuntia. Nämä tauot ohjelmassa antavat automatisoitavalle ohjelmistolle aikaa ladata uudelle sivulle, sillä robotti suorittaa usein nopeammin kuin ohjelmat ehtivät ladata, josta usein seuraa ongelmia, kun robotti yrittää suorittaa seuraavaa vaihetta latauksen vielä ollessa kesken. Seuraavaksi robotti painaa ’Import purchasing inforecord -nappia’ liikuttamalla hiirtä 150 pikseliä alas ja 150 pikseliä oikealle ja klikkaamalla vasemmalla hiiren napilla. Tällä tavalla päästään tiedostojen syöttämissivulle (Liite 14). Ohjelma odottaa puoli sekuntia ennen kuin se rupeaa etsimään seuraavaa bittikarttaa, joka on nimeltään ’Kello_uusi.PNG’. Tätä bittikarttaa käytetään usein ’ankkurina’ tässä näkymässä, sillä se on uniikki ja siten robotti löytää sen suhteellisen nopeasti. Kellopainikkeen päältä hiiri siirretään suhteellisesti ensin ’Purchasing organization’ -tekstikentän kohdalle ja sitten ’Plant’-tekstikentän kohdalle, joihin robotti myös kirjoittaa oleelliset arvot. Seuraavaksi robotti painaa ’Simple version -ruudun’ aktiiviseksi. Ylläkuvatut vaiheet tapahtuvat joka kerta oli muuttujan t arvo sitten 1, 2 tai 3.

Seuraavaksi määritellään aliohjelmat eri tiedostojen syöttämiselle. Eri tiedostot vaativat eri asetuksia tiedostoja syötettäessä ja oikeiden asetusten asettaminen riippuu aikaisemmin syötettyjen tiedostojen määrästä. Näistä riippuvaisuuksista johtuen eri tiedostojen syötöt päätettiin erottaa omiksi aliohjelmikseen, joiden toimintaa esitellään Kuvissa 23, 24 ja 25 tarkemmin.

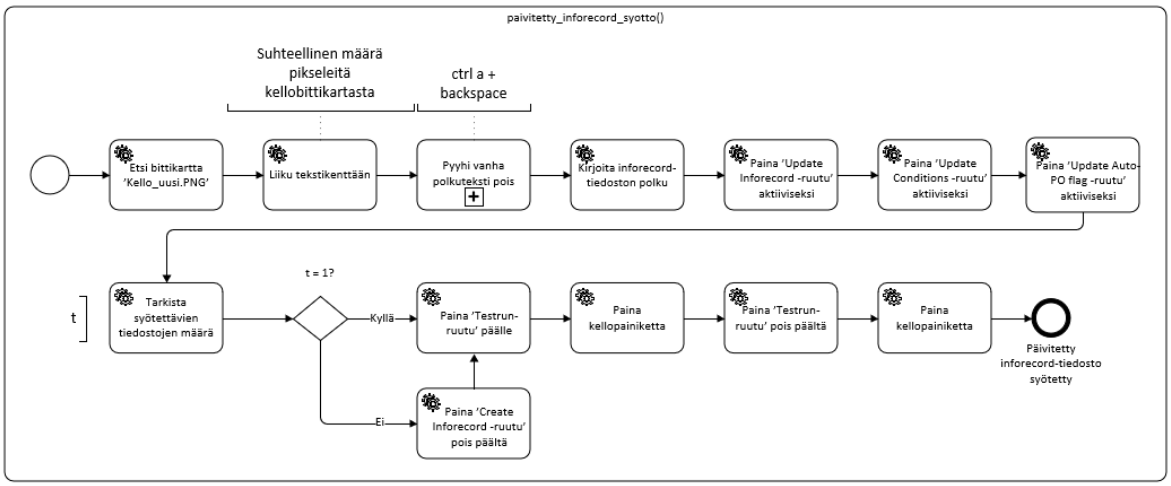


Kuva 23. Inforecord-tiedoston syöttö



Kuva 24. Plant-tason inforecord-tiedoston syöttö

Inforecordin ja plant-tason inforecordin syöttäminen vaativat samanlaisia asetuksia, eli ruudun ’Create Inforecord’ on oltava aktiivisena ja polku tiedostoon on oltava syötettynä oikein. Tämän jälkeen asetetaan ruutu ’Testrun’ aktiiviseksi ja ajetaan painamalla kellosta. Ohjelma odottaa testiajon loppuun. Sitten ’Testrun-ruutu’ otetaan pois päältä ja tiedot ajetaan oikeasti painamalla kelloa uudelleen ja odottamalla. Aliohjelma plant_inforecord_syotto ei eroa kauheasti edeltävästä aliohjelmasta. Erona on mahdollisen edellisen polun poistaminen ’File source -tekstikentästä’ ennen uuden polun kirjoittamista ja yksi ehtolauseke, joka testaa muuttujan t arvoa. Jos syötettäviä tiedostoja on 1 tai 2, ei tavallista inforecordia tarvitse syöttää, jolloin aliohjelmaa inforecord_syotto ei ajeta, jolloin ’Create Inforecord -ruutu’ ei ole valmiiksi valittuna ja se on tehtävä tässä kohdassa. Koska suurin osa hiiren liikkeistä perustuu suhteellisiin sijainteihin, on ’Testrun-ruudun’ sijainti myös määriteltävä uudelleen tässä tapauksessa, jonka jälkeen ohjelma jatkaa normaalisti ensin testaukseen ja sitten varsinaiseen ajoon kuten aikaisemmassa aliohjelmassa.

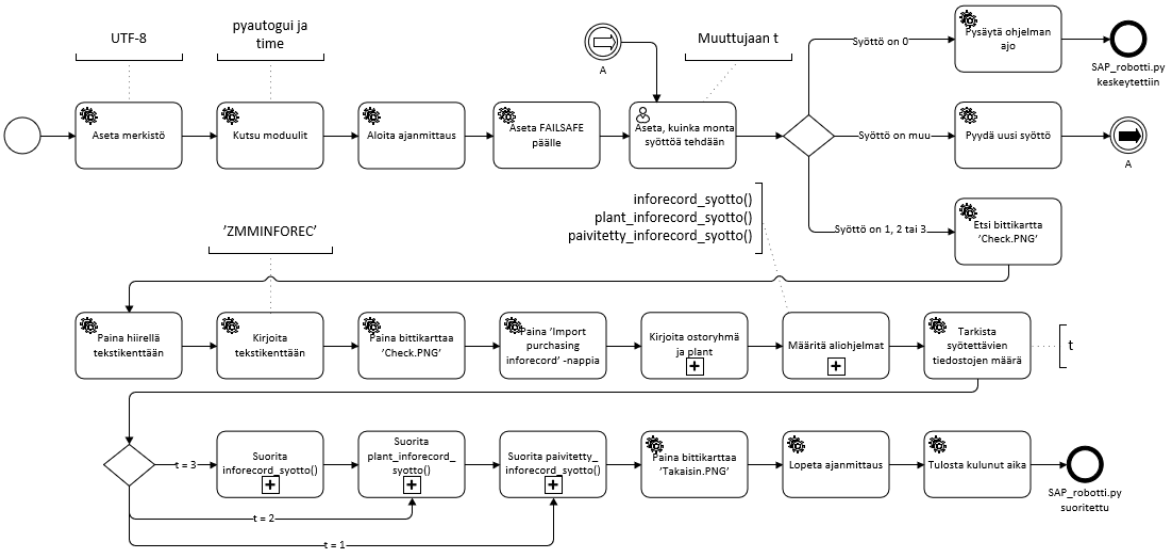


Kuva 25. Päivitetyn inforecord-tiedoston syöttö

Viimeinen aliohjelma määritellään seuraavaksi ja se eroaa enemmän kahdesta aikaisemmasta aliohjelmasta. Mennäkseen läpi tässä syötössä on oltava oikea polku päivitetyn inforecordin tiedostoon sekä ruutujen ’Update Inforecord’, ’Update Conditions’ ja ’Update Auto-PO flag’ on oltava aktiivisena. Sen sijaan ruudun ’Create Inforecord’ on oltava

epäaktiivisena, joten sen tila on tarkastettava ennen kuin ajot tehdään. Koska sekä inforecordin että plant-tason inforecordin syöttäminen vaativat ruudun ’Create Inforecord’ olevan aktiivinen, ainoa tilanne, jossa se ei ole jo aktiivinen on, kun muuttujan t arvo on 1. Tässä tapauksessa sen yli voidaan hypätä suoraan testaukseen ja varsinaiseen ajoon. Jos taas muuttujan t arvo on 2 tai 3, tarkoittaa se sitä, että ruutu ’Create Inforecord’ on muutettava epäaktiiviseksi ennen kuin siirrytään testaukseen ja varsinaiseen ajoon. Hiiren suhteellisten liikkeiden takia tässäkin on määriteltävä kaksi eri polkua ’Testrun-ruudun’ sijainnille.

Nyt kun kaikki tieto käyttäjältä on kerätty ja kaikki aliohjelmat on määritelty, voidaan ohjelman kulku määritellä eri tilanteissa. Koko ohjelman logiikka voidaan nähdä tiivistettynä Kuvassa 26.



Kuva 26. SAP_robotti.py tiivistettynä

Jos muuttuja t, eli syötettävien tiedostojen määrä on kolme, kaikki aliohjelmat kutsutaan vuorotellen ja robotti syöttää kaikki tiedostot. Jos taas tehdään vain kaksi syöttöä, kutsutaan vain aliohjelmat plant_inforecord_syotto ja paivitetty_inforecord_syotto. Lopuksi, jos vain yksi tiedosto syötetään, kutsutaan vain paivitetty_inforecord_syotto. Tämän jälkeen robotti palauttaa näkymän takaisin etusivulle etsimällä bittikartan ’Takaisin.PNG’. Ohjelma päättyy suoritusajan tulostuksella.

5.3 Suorituskyvyn muutoksen arviointi

Työn lähtökohtana oli tarjous-hankintaprosessin toimihenkilötyön suorituskyvyn parantaminen, joten tässä osiossa arvioidaan, miten työn ratkaisut vaikuttavat valittuun prosessin osaan. On pidettävä mielessä, että diplomityö on projektina suhteellisen lyhyt, joten tarkkoja pidemmän aikavälin mittauksia ei ehditä tekemään, vaan ne on määriteltävä pidemmällä ajanjaksolla diplomityön jälkeen. Kuten aikaisemmassa luvussa ”2.2 Prosessin suorituskyvyn mittaaminen” käsiteltiin, suorituskykyä voidaan mitata useassa eri muodossa. Tässä diplomityössä oltiin kiinnostuneita erityisesti ajallisen, taloudellisen ja laadullisen suorituskyvyn parantamisesta. Ajallinen ja taloudellinen suorituskyky ovat tässä tapauksessa erittäin sitoutuneita toisiinsa, sillä nopeampi nimikkeiden osto parantaa koko tuotantoketjun suorituskykyä, joka tuo mukanaan tietysti taloudellisia etuja. Myös pienemmällä mittakaavalla ostajien työn pitäisi tehostua, kun he voivat käyttää aikaansa harkintaa vaativiin ostoihin rutiiniostojen, tiedonhaun ja rutiininomaisten syöttöjen sijaan. Taulukkoon 11 on koottu kaikkien ohjelmien mitatut kestot erilaisissa tilanteissa. Ajat mitattiin peräjälkeen useaan kertaan ja samankaltaisissa olosuhteissa.

Taulukko 11. Ohjelmien suoritusten aikahaarukat

Ohjelma	**Automaation kesto**
Luo_Excelit.py	0,5–0,6 sekuntia
Haku.py	15–180 sekuntia
Hankintasuunnitelma optimaalisessa muodossa	15–20 sekuntia
Hankintasuunnitelma epäoptimaalisessa muodossa	~ 180 sekuntia
SAP_robotti.py	32–58 sekuntia
Yhdellä tiedostolla	32–38 sekuntia
Kahdella tiedostolla	45–53 sekuntia
Kolmella tiedostolla	54–58 sekuntia

Luo_Excelit.py ohjelma suorittaa noin puolessa sekunnissa, jos sen ei tarvitse hakea tietoa hankintasunnitelmasta. Jos hankintasuunnitelma on avattava, lisätään Haku.py ohjelmassa kulunut aika Excelien luontiin kuluvaan aikaan. Hankintasuunnitelman muoto vaikuttaa huomattavasti ohjelman suoritusaikaan. Turhasta muotoilusta riisutussa muodossa se

suorittaa keskimäärin noin 17 sekunnissa, mutta epäoptimaalisessa muodossa suorituksessa voi mennä jopa kolme minuuttia. Suurin osa tästä ajasta kuluu vain tiedoston avaamiseen ensinnäkin, koska siinä on niin paljon rivejä usealla sivulla ja toisekseen, koska siinä on erittäin paljon raskasta muotoilua. Jatkoa ajatellen tämän tiedoston tehostaminen olisi suositeltavaa, sillä se nopeuttaisi sekä tätä automaatiota että suunnittelijoiden ja ostajien työtä, joiden on avattava tämä tiedosto lähes päivittäin. Perustilanne kuitenkin on, ettei hakuohjelmaa tarvitse kutsua ja vaikka tarvitsisikin, on suoritus myös epäoptimaalisessa muodossa nopeampi kuin tiedonhakeminen ja kirjoittaminen käsin. Lisäksi, koska ohjelma toimii taustalla eikä vaadi graafista käyttöliittymää, hiirtä tai näppäimistöä, käyttäjä voi tehdä jotain muuta suorituksen aikana, kunhan hän ei vain avaa ostojonon tai hankintasuunnitelman tiedostoja.

Esitetyt ratkaisut vaikuttavat suoraan oston tarvitseman palveluajan lyhentämiseen ja pidemmällä aikavälillä myös odotusajan lyhentämiseen, kun tilaukset saadaan nopeammin ja varmemmin tehtyä. Tällä on vaikutusta vähintään muuttuviin kustannuksiin, kun seisokkien ja odottelun määrä vähenee. Ratkaisulla odotetaan olevan positiivisia vaikutuksia sisäiseen laatuun ostajien työn helpottuessa ja kiireen alla tehtävien päätösten vähentyessä. Automaatio tuo myös parempaa joustavuutta prosessin osaan, kun suuretkin ostojonot voidaan purkaa sekunneissa. Joustavuutta ei työn suunnittelussa määritelty seurattavaksi suorituskyvyksi, mutta joustavuudella on vaikutusta sekä ajalliseen, taloudelliseen että laadulliseen suorituskykyyn, mistä syystä sitä päätettiin kevyesti käsitellä myös tässä.

Myös SAP-robotin suoritusaikoja mitattiin eri tilanteissa. Suorituksessa menee noin puolesta minuutista yhteen minuuttiin riippuen siitä, kuinka monta tiedostoa syötetään. Jokainen näistä variaatioista oli nopeampi kuin vastaava käyttäjän syöttönopeus tavallisessa tilanteessa. Kuten Taulukosta 12 voidaan nähdä, absoluuttinen ja suhteellinen ero käsin syötettyjen ja ohjelmallisesti syötettyjen aikojen välillä kasvaa huomattavasti, mitä pidemmästä suorituksesta on kyse. Yhden tiedoston syötön automaatio on kyseenalaista, kun verrataan sen rakennuksen vaivaa siitä saataviin hyötyihin, mutta jo kahden ja varsinkin kolmen tiedoston kohdalla nähdään huomattavaa ajansäästöä. Eteenkin pitkissä SAP-syötöissä siis ohjelmistorobotista saataisiin huomattavin hyöty.

Taulukko 12. Vertailu käsin ja ohjelmallisesti syötettävien aikojen välillä

Syötettävien tiedostojen määrä	**Aika käsin (sekuntia)**	**Aika ohjelmallisesti (sekuntia)**	**Absoluuttinen ero (sekuntia)**	**Suhteellinen ero (%)**
1	~ 40	~ 36	4	~ 10
2	~ 60	~ 49	11	~ 18
3	~ 90	~ 56	34	~ 38

Ei pidä unohtaa, että automaatiosta ja robotista saadun ajallisen ja sitä kautta myös taloudellisen suorituskyvyn parannuksen lisäksi, myös laadullinen suorituskyky saattaa parantua ratkaisun myötä. Koska automaatio ja ohjelmistorobotit suorittavat niille määritellyt työt aina samalla tavalla, tulisi syöttövirheiden määrän vähentyä verrattuna ihmisen käsin tekemään rutiinisyöttöön. Tietysti tämä edellyttää sitä, että ohjelmia ylläpidetään ja päivitetään toimintatapojen tai ohjelmistojen muuttuessa. Huonompana puolena robotissa on se, että ohjelman suoritus vaatii tietyn näkymän tietokoneen näytölle, eikä käyttäjä voi käyttää hiirtä ja näppäimistöä ajon aikana lähtötietojen syöttämisen jälkeen. Ajossa ei kuitenkaan mene edes minuuttia ja tulevaisuudessa, jos robotteja ruvetaan käyttämään yrityksessä enemmän, voidaan ajot hoitaa etäyhteydellä pelkästään robottien ajoon tarkoitetulla tietokoneella tai virtuaalikoneella, jolloin robotin suoritus voidaan jättää taustalle.

Aikaisemmin käsiteltiin vain yksittäisiä ajoja ja sekunteja, mutta jos harkitaan, että esimerkiksi vuositasolla seitsemän ostajan käyttäessä tiedonsyöttöön robottia päivittäin ja työpäiviä ollessa 250, niin aikaa säästyy suoraan keskimäärin noin yhden henkilötyöpäivän verran. Tämä ei kuitenkaan ota huomioon Excelien automaattisessa luonnissa säästettyä aikaa taikka kaskadiefektiä siitä, että ostajat voivat keskittyä nopeammin harkinnanvaraisiin ostoihin ja siten koko hankintaprosessi tehostuu. Kuten aikaisemmassa luvussa ”2.3 Toimihenkilötyön määritelmä ja mittaus” todettiin, toimihenkilötyön mittaaminen on haastavaa eikä jokaiseen tilanteeseen löydy täysin kattavaa tai objektiivista mittaristoa. Tämän työn osalta olisi edullista seurata pidemmällä aikavälillä automaattisten ostojen määrään kehitystä ja verrata sitä vastaaviin lukuihin ennen automaatioratkaisun käyttöönottoa. Tällä tavalla saataisiin mitattua ratkaisun toivottua vaikutusta suoraan. Toimihenkilötyön suorituskyvyn ajallista tehostumista voidaan myös arvioida tästä muutoksesta, mutta on pidettävä mielessä, ettei se ole

täysin tarkka arvio. Ostajat tulisi arvioida yhtenä ryhmänä tässä tapauksessa, sillä yksittäisen toimihenkilön suorituskyvyn arvioiminen on hyvin haastavaa, epätarkkaa eikä siten kovin hyödyllistä.

6 Johtopäätökset

Tässä luvussa vedetään työn johtopäätökset yhteen vastaamalla johdannossa asetettuihin tutkimuskysymyksiin ja pohtimalla, miten tässä työssä aloitettua tutkimusta voitaisiin hyödyntää ja kehittää eteenpäin. Luvussa avataan myös tutkimuksen rajoitteita ja ratkaisun jatkokehitystarpeita.

6.1 Vastaukset tutkimuskysymyksiin

Miten toimihenkilötyön suorituskykyä voidaan parantaa tarjous-hankintaprosessissa?

Toimihenkilötyön tehostamiselle löydettiin useita erilaisia keinoja tarjous-hankintaprosessin piiristä. Tehostuskeinot liittyivät erityisesti tiedonhakuun ja -siirtämiseen sekä kommunikaatioon ulkopuolisten sidosryhmien kanssa. Prosessimallinnus oli avainasemassa muun muassa näiden tehostustoimien löytämisessä, joten sitä avataan seuraavaksi. Tämän jälkeen tarkastellaan tätä tutkimuskysymystä myös prosessiautomaation sekä prosessimallinnuksen ja -automaation yhteisvaikutuksen näkökulmista.

Miten prosessimallinnus voi tehostaa toimihenkilötyötä?

Mallinnuksen kautta saatiin selkeämpi kuva prosessin nykytilasta ja myös prosessien asiantuntijat kokivat koko prosessin näkemisen alusta loppuun hyödylliseksi. Selkeällä tilannekuvalla saatiin pullonkaulat, odotusajat ja muut epäkohdat helposti esille, mikä auttoi erityisesti automatisoitavan prosessin osan tunnistuksessa. Mallinnuksessa myös tiedonkulku saatiin selkeästi esille, mikä auttoi varsinkin automaation suunnittelussa, kun tietolähteet ja -tarpeet olivat selvillä. Tiedonkulun mallinnus toi myös esille, kuinka monesta eri paikasta tietoa on haettava tai toisin sanoen, kuinka monessa eri paikassa hyödyllistä tietoa on yrityksessä tallennettuna. Tiedon saatavuus ja ajankohtaisuus paljastuivat yleensä kriittisimmiksi tekijöiksi eri osaprosesseissa, joten tämä tiedon levinneisyys tulisi käsitellä yrityksessä mahdollisimman nopeasti. Tulokset mallinnuksen tuomista eduista, kuten hiljaisen tiedon muuttaminen näkyväksi sekä toiminnan ja tiedonkulun selkeästä esittämisestä ovat johdonmukaisia aikaisemmin esitetyn kirjallisuuden kanssa (Ahmad & van Looy 2020; Dumas et al. 2013, 16–23; Luukkonen et al. 2012).

Miten prosessiautomaatio hyödyttää toimihenkilötyötä?

Prosessiautomaatiolla saatiin prosessin osan palveluaikaa lyhyemmäksi, jolla on suoria vaikutuksia myös prosessin muuttuviin kustannuksiin. Ostajat pystyvät automaation avulla keskittymään ainoastaan inhimillistä harkintaa vaativiin ostoihin, joka taas voi lyhentää odotusaikaa myöhemmin valmistusprosessin ketjussa, kun ostotilaukset saadaan tehtyä nopeammin. Sen lisäksi, että tiedon etsintään ja syöttämiseen kuluva aika lyhenee myös syöttöjen laadun pitäisi nousta, koska ohjelmat kopioivat ja liittävät tiedot aina samalla tavalla. Automaatioratkaisu onnistui suunnitelman mukaisesti, eli kaikki halutut vaiheet saatiin toteutettua poikkeuskäsittelyineen. Pidemmän aikavälin tavoitteita tulee seurata välikatsauksilla käyttäjien kanssa ja ylläpitoa on tehtävä mahdollisten ongelmien tai muutosten ilmetessä. Tulokset ovat yhdenmukaiset prosessiautomaatiosta esitettyjen teoreettisten väitteiden, kuten automaatiolle hyvin soveltuvien tehtävien valinnan, automaatioon liittyvien haasteiden ja ohjelmistorobotteihin liittyvien lisähuomioiden kanssa (Hyun et al. 2021; Santos et al. 2020; Wellman et al. 2020; Fung 2014; Jimenez-Ramirez et al. 2019; Wanner et al. 2019; Yatskiv et al. 2019; Leopold et al. 2018; Kasslin 2017).

Miten prosessimallinnus ja -automaatio voivat tukea toisiaan prosessin suorituskyvyn parannuksessa?

Tämän työn pohjalta voitaisiin sanoa prosessimallinnuksen ja -automaation toimivan erittäin hyvin yhdessä, mikä on johdonmukaista aiemman tutkimuksen kanssa (Dumas et al. 2013, 6; Paschek et al. 2017; Kirchmer & Franz 2019; Kroll et al. 2016; Signavio Whitepaper 2019; Tornbohm 2017). Automatisoitavan prosessin osan löytäminen ja harkinta helpottuvat huomattavasti mallinnuksen kautta ja automaation toteutus helpottui huomattavasti, kun tiedonkulku ja prosessin logiikka olivat selvillä ja helposti luettavissa. Varsinkin ohjelmistorobottien rakentaminen vaatii hyvinkin yksityiskohtaista prosessimallinnusta ja tiedonkulun selvitystä, joten sen tueksi voitaisiin erityisesti suositella prosessimallinnusta. Automaation päivittäminen ja ajankohtaisena pitäminen voi myös helpottua, kunhan mallinnusta pidetään myös systemaattisesti ajan tasalla. Mallinnus tukee siis automaation rakennusta, joka taas tukee prosessin ajallista, taloudellista, laadullista ja joustavuuden suorituskykyä.

6.2 Tulosten hyödynnettävyys ja rajoitteet

Ratkaisua voidaan lähes välittömästi hyödyntää siinä prosessin osassa, johon se on rakennettu. Käyttäjät on ohjeistettava työkalun käytöstä ja käytössä ilmenevät viat on vielä korjattava sekä ohjelmien yleisestä ylläpidosta on huolehdittava. Ratkaisua todennäköisesti voitaisiin käyttää kevyesti muokattuna muissakin organisaation ostofunktioissa, sillä ostoprosessit ovat suhteellisen vakioituja, mutta tämä vaatii lisäselvitystä, joka menee laajuusalueeltaan tämän diplomityön ulkopuolelle. Ratkaisun suora hyödynnettävyys muissa samana alan yrityksissä riippuu suuresti niistä työkaluista, joita niissä käytetään. Kuitenkin ydinidea prosessimallinnuksen ja -automaation synergian tuomista eduista on hyödynnettävissä missä tahansa teknologia- ja teollisuusyrityksessä. Prosessimallinnus on automaation kehittämistä ja toimihenkilötyön tehostamista helpottava työkalu, jota yrityksissä tulisi hyödyntää enemmän.

Rajausta SAP:in ulkopuolelle ei voitu tehdä, vaikka se työn alussa määriteltiin rajoitteeksi. Mallinnusta tehdessä kävi selväksi, että lähes kaikki tarjous-hankintaprosessin osat koskettavat SAP-rajapintaa ainakin jossain kohtaa. Tästä syystä olisi suositeltavaa jatkokehitystä varten, että luvat SAP-ympäristön automaation ja ohjelmalliseen käsittelyyn selvitettäisiin. Erityisesti pitkät SAP-syötöt saataisiin huomattavasti nopeammiksi ja virheettömimmiksi ohjelmistoroboteilla. Valitettavasti koko tiedonhaun prosessin osaa ei voitu automatisoida, mutta tästä pienestä prosessin osan automaatiosta voitiin kuitenkin oppia paljon. Suurin rajoitus automaatioratkaisulle oli tietenkin SAP, mutta myös muita pienempiä rajoituksia löytyi. Nyt ostajan on tallennettava ostojono tiettyyn paikkaan tietyssä muodossa, joka voitaisiin myös tehdä automaattisesti, mutta ostajan on pitänyt tehdä se jo joka tapauksessa ja automatisaatioon menevä vaiva olisi todennäköisesti suurempi kuin siitä saatu hyöty. Suurempi ongelma on hankintasuunnitelman latauksen kanssa. Koska sitä päivitetään jatkuvasti, tulisi ostajan ladata se uudestaan joka kerta, kun ohjelma ajetaan. Hankintasuunnitelma on myös saatavilla yrityksen käyttämällä tiedonjakoalustalla, mutta se on tallennettuna siellä binääritaulukon muodossa, jota OpenPyXL-moduulin työkalut eivät tue. Tämän lisäksi taulukko on suuri, monisivuinen ja täynnä raskasta muotoilua, mikä tekee sen käsittelystä suhteellisen hidasta myös ohjelmallisesti. Tähän ongelmaan voidaan kuitenkin etsiä ratkaisu, joko tehostamalla alkuperäistä taulukkoa tai tekemällä alkuperäiseen taulukkoon perustuva

linkitetty kopio, joka pysyy ajan tasalla ja on valmiina kevyemmässä muodossa semmoisessa paikassa, josta ohjelma voi hakea sen.

6.3 Jatkokehityskohteet

Seuraavaksi käydään joitain jatkokehitysideoita perustuen diplomityön tuloksiin. Ensinnäkin lupa hyödyntää ohjelmistorobotteja SAP-ympäristössä olisi pitkällä aikavälillä hyvin oleellinen parannus varsinkin, jos ruvetaan toteuttamaan muita tarjous-hankintaprosessin toimihenkilötyön tehostustoimia tai esimerkiksi talouspuolen toimia. Lisäksi diplomityössä rakennetun automaatiodemon pohjana käytettyä automaattista ostamista voitaisiin laajentaa muihinkin perustavaran toimittajiin. Asiantuntijahaastatteluiden perusteella tarjolla olisi ainakin muutama sopiva vaihtoehto tällaiseen laajennokseen. Se lisäisi myös rakennetun automaation käyttöä, kun useampi nimike menisi automaattiseen ostoon ja lisätietoa tarvitsisi hakea useammin. Yksi luonteva jatkokehityskohde rakennetulle automaatiolle olisi piirustusten ja dokumenttien automaattinen liittäminen ostoihin. Ostaja joutuu ne tällä hetkellä lähettämään käsin, vaikka itse tilaus lähtisikin automaattisesti. Tähän jatkokehitykseen on jo luotu pohja, sillä hankintasuunnitelmasta saadaan ainakin osa tarvittavien dokumenttien nimistä. Dokumenttien hakeminen ja liittäminen sähköpostiin onnistuu Pythonilla suhteellisen helposti. Jatkokehityksen tai -tutkimuksen kannalta olisi myös mielenkiintoista pyrkiä viemään prosessimallinnuksen ja -automaation synergiaa vielä eteenpäin yhdistämällä mallinnus ja automaatio siten, että mallinnuksen parametrien muuttaminen muuttaisi myös automaation toimintaa. Tällä tavalla sykli prosessimallinnuksen ja prosessiautomaation kanssa saataisiin suljettua työn teoreettisen viitekehyksen osoittamalla tavalla. Tässä voitaisiin esimerkiksi yhdistää MS Visio ja MS Power Automate tai vastaava yhdessä toimiva ohjelmistopari.

Suurempana jatkokehitysprojektina voisi tämän diplomityön pohjalta ehdottaa kokonaisvaltaista tiedostojen hallintajärjestelmää, joka ulottuu sekä yrityksen sisäisiin funktioihin että ulkoisille toimijoille. Tässäkin työssä rakennetun automaatiodemon kaltaiset ratkaisut ovat oikeastaan vain laastareita, jotka pyrkivät tehostamaan sirpaleisen tiedon käsittelyä. Yrityksessä on tällä hetkellä paljon irtonaisia tietolähteitä, jotka eivät välttämättä kommunikoi keskenään kitkattomasti. Mallinnuksen yhteydessä tehdyissä toimihenkilöiden haastatteluissa

kävi ilmi, kuinka monesta eri lähteestä tietoa on haettava ja kuinka kriittistä ajankohtaisen tiedon saanti on. Myös ajankohtaisen tiedon jakaminen ja saaminen toimittajilta ja asiakkailta on erittäin tärkeää sekä ajallisen että laadullisen suorituskyvyn kannalta. Pitkällä aikavälillä tällaisen systemaattisen tiedonkokoamiseen ja -jakamiseen tarkoitetun järjestelmän hankkiminen on välttämätöntä, jos toimihenkilöiden työtä halutaan tehostaa merkittävällä tavalla. Yleisenä tutkimuksellisena jatkokehityksenä tämän diplomityön pohjalta voitaisiin ehdottaa samankaltaisen tutkimuksen tekemistä erilaisissa yrityksissä ja/tai erilaiseen liiketoimintaprosessin osaan. Tämän diplomityön ratkaisu on hyvin rajoitettu sen kontekstiin, mutta yleisen mallin tai kaavan rakentaminen voisi olla yksi mielenkiintoinen jatkokehityskohde.

7 Yhteenveto

Digitalisaation, kansainvälistyvien markkinoiden ja tiukentuvan kilpailun takia yritysten on löydettävä uusia tapoja toimintansa tehostamiseksi. Eteenkin teknologia- ja teollisuusyrityksissä työ on muuttunut jatkuvasti enemmän asiantuntijapainotteiseksi työksi, jolloin perinteisemmät suorittavan työn tehostustoimet eivät enää riitä. Tästä syystä moni yritys on kiinnostunut erilaisista toimihenkilötyön tehostamisen keinoista kuten rutiinien automatisaatiosta ja ohjelmistoroboteista.

Työn tavoitteena oli mallintaa ANDRITZ Savonlinna Works Oy:n tarjous-hankintaprosessi mahdollisimman todenmukaisessa tilassa, paikantaa siihen perustuen mahdollisia automaatiokohteita ja toteuttaa prosessin osan suorituskykyä parantava automaatiodemo. Tämän pohjalta saatiin konkreettinen tulos yrityksen prosessin tehostamiseksi ja samalla saatiin tutkittua prosessimallinnuksen ja -automaation synergiaa. Tämä toteutettiin luomalla ensin työlle teoreettinen tausta, jossa prosessimallinnus ja -automaatio yhdistettiin perustuen prosessijohtamisen elinkaareen. Lisäksi määriteltiin automatisoitavan prosessin osan kriteeristö sekä työssä käytetty mallinnusnotaatio, eli BPMN. Seuraavaksi ruvettiin rakentamaan malleja ohjausryhmän kanssa käytyjen palaverien ja osaprosessien asiantuntijoiden haastatteluiden pohjalta. Kun kaikki mallit oli validoitu, niitä ruvettiin analysoimaan mahdollisten automaatiokohteiden tunnistamiseksi. Parhaita vaihtoehtoja vertailtiin perustuen aikaisemmin luotuun kriteeristöön sekä diplomityön tavoitteisiin ja lopulta valittiin prosessin osa, jossa puuttuvia tietoja täytetään automaattisia ostoja varten. Koska tämä prosessi on kosketuksissa SAP-rajapinnan kanssa, tuli se ottaa huomioon toteutuksen suunnittelussa. Automaatiodemon käyttämät ohjelmat rakennettiin Python-ohjelmointikielellä erityisesti hyödyntäen OpenPyXL- ja PyAutoGUI-moduuleja.

Tuloksena syntyi siis todenmukainen mallinnus yrityksen tarjous-hankintaprosessista sekä siinä liikkuvasta tiedosta BPMN-syntaksilla. Tämän pohjalta rakennettiin automaatiodemo, joka koostuu kolmesta ohjelmasta. Ensimmäinen ohjelma luo, muotoilee ja täyttää kolme Excel-tiedostoa perustuen syötettyyn ostojonoon siten, että käyttäjän on vain ladattava ne

SAP:iin, jotta automaattinen osto-ohjelma saa kaikki tarvitsemansa tiedot ja menisi läpi. Jos tiettyjä arvoja puuttuu, niin ensimmäinen ohjelma kutsuu toista ohjelmaa, joka hakee puuttuvat tiedot yrityksen hankintasuunnitelmasta perustuen nimikkeen työnumeroon. Kolmas ohjelma on ohjelmistorobotti, joka tehtiin simuloituun SAP-ympäristöön osoittamaan, kuinka paljon tehokkaampia tämä ratkaisu ja potentiaaliset tulevaisuuden ratkaisut voisivat olla, jos SAP-ympäristöä voitaisiin automatisoida. Prosessin osan laadullisen, ajallisen, taloudellisen ja joustavuuden suorituskyvyn arvioitiin parantuvan tämän ratkaisun käyttöönotosta, mutta pidemmän aikavälin seurantaa tarvitaan näiden arvioiden varmistamiseksi.

Työhön perustuvien johtopäätösten mukaan prosessimallinnus ja -automaatio tukevat toisiaan useilla tavoilla toimihenkilötyön tehostamisessa tarjous-hankintaprosessissa. Automatisoitavan prosessin osan löytäminen helpottui huomattavasti, kun koko prosessin kulku ja konteksti oli saatavilla yhdessä paikassa. Tällä tavalla pystyttiin löytämään useita potentiaalisia kohteita, mutta automaatiokriteeristöstä oli myös apua lopullisessa prosessin osan valinnassa. Myös itse automaation toteutus helpottui, kun prosessin tiedonkulku ja prosessilogiikka olivat selvillä. Erityisesti ohjelmistorobottien rakennuksessa on mentävä hyvinkin yksityiskohtaiselle tasolle prosessin kuvauksessa, jotta se saadaan toimimaan toivotulla tavalla. Prosessimallinnus voi myös tukea automaation ylläpitoa siinä tapauksessa, että mallia pidetään systemaattisesti ajan tasalla. Jos automaatio tarvitsee päivitystä tai se toimii virheellisesti, voidaan sitä vastaava malli avata ja nähdä, miten sen kuuluisi suorittaa ja mitä tietoa se käyttää. Tulevaisuuden kannalta yrityksen kannattaisi harkita SAP-ympäristön automaation rajoitteiden selvittämistä, sillä suurin osa sen prosesseista koskettaa SAP-rajapintaa ainakin jossain kohdassa. Työssä rakennettua automaatiodemoa voitaisiin kehittää eteenpäin siten, että se hakisi ja lähettäisi automaattisesti toimittajien vaatimat piirustukset ja muut dokumentit. Automaatiodemon vaikutusta ja yleisesti automaatioastetta yrityksessä voitaisiin nostaa, jos useampi perusosien toimittaja tuotaisiin automaattisen oston piiriin. Suurempana jatkokehitysprojektina yritykselle ehdotettiin kokonaisvaltaista tiedostojen hallintajärjestelmää, joka ulottuu sekä yrityksen sisäisiin funktioihin että ulkoisille toimijoille. Tiedonkulku yrityksen sisällä sekä toimittajien ja asiakkaiden kanssa on yksi kriittisimmistä ajallisen, laadullisen ja taloudellisen suorituskyvyn määrittelijöistä tarjous-hankintaprosessin jokaisessa osaprosessissa.

Lähdeluettelo

Ahmad, T. & van Looy, A. 2020. Business Process Management and Digital Innovations: A Systematic Literature Review. Sustainability. Vol. 12(17). Saatavilla: https://doi.org/10.3390/su12176827

Ahmed, N.U., Ma, C.S. & Montagno, R.V. 1991. Measuring White-Collar Productivity. American Journal of Business. Vol. 6(1). Saatavilla: https://www.proquest.com/docview/1012242552

Amaratunga, D. & Baldry, D. 2002. Moving from performance measurement to performance management. Facilities. Bradford, West Yorkshire, England. Vol. 20(5/6), 217–223. Saatavilla: https://doi.org/10.1108/02632770210426701

ANDRITZ. 2023. ANDRITZ Savonlinna Works Oy, Savonlinna. [Yrityksen esittelysivu]. Saatavilla: https://www.andritz.com/pulp-and-paper-en/locations/savonlinna-works-oy/savonlinna-works-oy-fi

Asatiani, A. & Penttinen, E. 2016. Turning robotic process automation into commercial success - Case OpusCapita. Journal of Information Technology Teaching Cases. Vol. 6(2), 67–74. Saatavilla: https://www.researchgate.net/publication/303460189_Turning_robotic_process_automation_into_commercial_success_-_Case_OpusCapita

Bradford, L. & Dumas, M. 2007. Getting Started with YAWL. Yawl Foundation. Saatavilla: https://www.researchgate.net/publication/265352090_Getting_Started_with_YAWL

Dumas, M., La Rosa, M., Mendling, J. & Reijers, H.A. 2013. Introduction to Business Process Management; Fundamentals of Business Process Management. 2nd edition Springer, Berlin, Germany, ISBN 9783642331428. Saatavilla: https://dl.uswr.ac.ir/bitstream/Hannan/141382/2/9783662565087.pdf

Ferraris, A., Monge, F. & Mueller, J. 2018. Ambidextrous IT capabilities and business process performance: An empirical analysis. Bus. Process Manag. J. Vol. 24, 1091–1109. Saatavilla: https://www.semanticscholar.org/paper/Ambidextrous-IT-capabilities-and-business-process-Ferraris-Monge/fddbdeaf46c5043e2ac0e2346fd8339f200d34d3

Fung, H.P. 2014. Criteria, use cases and effects of information technology process automation (ITPA). Advanced Robotics & Automation. Vol. 3(3), 1-10. Saatavilla: https://www.researchgate.net/publication/274378872_Criteria_Use_Cases_and_Effects_of_Information_Technology_Process_Automation_ITPA

Hyun, Y., Lee, D., Chae, U., Ko, J. & Lee, J. 2021. Improvement of Business Productivity by Applying Robotic Process Automation. Applied Sciences. Vol. 11(22). Saatavilla: https://doi.org/10.3390/app112210656

Ilmarinen, V. & Koskela, K. 2015. Digitalisaatio : yritysjohdon käsikirja. Helsinki: Talentum, Finland. ISBN: 978–952–14–2628–5.

Jimenez-Ramirez, A., Reijers, H.A., Barba, I. & Del Valle, C. 2019. A method to improve the early stages of the robotic process automation lifecycle. Springer. Giorgini, P., Weber, B. (eds.) Advanced Information Systems Engineering. CAiSE 2019. LNCS, Vol. 11483, 446–461. Saatavilla: https://doi.org/10.1007/978-3-030-21290-2_28

Kasslin, H. 2017. Heavyweight and Lightweight Process Automation - How Do Companies Select between RPA and Back-End Automation? Diplomityö, Aalto University. Saatavilla: https://aaltodoc.aalto.fi:443/handle/123456789/26827

Kinney, M.R. & Rayborn C.A. 2009. Cost Accounting, foundations and evolutions. Thomson Higher Education, 7th edition, Mason, Ohio, USA. ISBN 9780324560558.

Kirchmer, M. 2017. Robotic Process Automation - Pragmatic Solution or Dangerous Illusion? Business Transformation and Operational Excellence Summit Insights. Saatavilla: https://www.researchgate.net/publication/317730848_Robotic_Process_Automation_-_Pragmatic_Solution_or_Dangerous_Illusion

Kirchmer, M. & Franz, P. 2019. Value-driven Robotic Process Automation (RPA). Springer, Cham. Shishkov, B. (ed.) Business Modeling and Software Design. Vol. 356, 31–46. Saatavilla: https://doi.org/10.1007/978-3-030-24854-3_3

Kroll, C., Bujak, A., Darius, V., Enders, W. & Esser, M. 2016. Robotic process automation - robots conquer business processes in back offices. Study, Capgemini Consulting and Capgemini Business Services. Saatavilla: https://www.capgemini.com/consulting-de/wp-content/uploads/sites/32/2017/08/robotic-process-automation-study.pdf

König, M., Bein, L., Nikaj, A. & Weske, M. 2020. Integrating Robotic Process Automation into Business Process Management. Hasso Plattner Institute, University of Potsdam. Asatiani, A., et al. (Eds.) 2020. BPM Blockchain and RPA Forum 2020, LNBIP. Vol. 393, 132–146. Saatavilla: https://link-springer-com.ezproxy.cc.lut.fi/content/pdf/10.1007/978-3-030-58779-6.pdf?pdf=button

Lacity, M., Willcocks, L.P. & Craig, A. 2016. Robotic process automation at Telefonica O2. MIS Quarterly Executive. Vol. 15(2). Saatavilla: https://eprints.lse.ac.uk/64516/1/OUWRPS_15_02_published.pdf

Leopold, H., van der Aa, H. & Reijers, H.A. 2018. Identifying Candidate Tasks for Robotic Process Automation in Textual Process Descriptions. Enterprise, Business-Process and Information Systems Modeling. Vol. 318, 67–81. Saatavilla: https://research.vu.nl/en/publications/identifying-candidate-tasks-for-robotic-process-automation-in-tex

Luukkonen, I., Mykkänen, J., Itälä, T., Savolainen, S. & Tamminen, M. 2012. Toiminnan ja prosessien mallintaminen: Tasot, näkökulmat ja esimerkit. Itä-Suomen yliopisto, Kuopio. ISBN: 978–952–61–0687–7. Saatavilla: https://www.researchgate.net/publication/257528716_Toiminnan_ja_prosessien_mallintaminen_Tasot_nakokulmat_ja_esimerkit

Mendling, J. 2022. Advancing Business Process Science via the Co-evolution of Substantive and Methodological Knowledge. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds) Business Process Management. BPM 2022. Lecture Notes in Computer Science Vol. 13420. Springer, Cham. Saatavilla: https://doi-org.ezproxy.cc.lut.fi/10.1007/978-3-031-16103-2_1

Object Management Group. 2014. Business Process Model and Notation version 2.0.2. Saatavilla: https://www.omg.org/spec/BPMN/2.0.2/About-BPMN/

Object Management Group. 2017. United Modeling Language (UML) version 2.5.1. Saatavilla: https://www.omg.org/spec/UML/2.5.1

Paschek, D., Luminosu, C.T. & Draghici, A. 2017. Automated business process management - in times of digital transformation using machine learning or artificial intelligence. MATEC Web of Conferences, 121. Saatavilla: https://www.semanticscholar.org/paper/Automated-business-process-management-%E2%80%93-in-times-of-Paschek-Luminosu/33f2ca80e54bcd7f824810adc4ebb2caaa3eb938#citing-papers

Peña, J., Bravo, A. & Resinas, M. 2022. BPM in Digital Transformation: New Tools and Productivity Challenges. In: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (eds) Business Process Management. BPM 2022. Lecture Notes in Computer Science. Vol. 13420. Springer, Cham. Saatavilla: https://doi-org.ezproxy.cc.lut.fi/10.1007/978-3-031-16103-2_2

Rantanen, H. & Holtari, J. 1999. Yrityksen suorituskyvyn analysointi. Lappeenranta, Lappeenrannan teknillinen korkeakoulu, Tuotantotalouden osasto, Tutkimusraportti. ISSN 0784-7688. Saatavilla: https://docplayer.fi/1578427-Yrityksen-suorituskyvyn-analysointi.html

Read the Docs. 2023a. openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files. Kehittäjät: Eric Gazoni & Charlie Clark. Saatavilla: https://openpyxl.readthedocs.io/en/stable/

Read the Docs. 2023b. Welcome to PyAutoGUI's documentation! Kehittäjä: asweigart. Saatavilla: https://pyautogui.readthedocs.io/en/latest/#

Ruževičius, J., Milinavičiūtė, I. & Klimas, D. 2012. Peculiarities Of The Business Process Management Lifecycle At Different Maturity Levels: The Banking Sector's Case. Issues of Business and Law 4, no. 2012, 69-85. Saatavilla: https://web-s-ebscohost-com.ezproxy.cc.lut.fi/ehost/pdfviewer/pdfviewer?vid=0&sid=95f8994d-6830-466a-916d-966e9b71e3f9%40redis

Salminen, A. 2011. Mikä kirjallisuuskatsaus? : johdatus kirjallisuuskatsauksen tyyppeihin ja hallintotieteellisiin sovelluksiin. Vaasan yliopisto, opetusjulkaisuja 62, julkisjohtaminen 4. Saatavilla: https://www.uwasa.fi/materiaali/pdf/isbn_978-952-476-349-3.pdf

Santos, F., Pereira, R. & Vasconcelos, J.B. 2020. Towards robotic process automation implementation: an end-to-end perspective. Business Process Management Journal. Vol. 26(2), 405-420. Saatavilla: https://doi.org/10.1108/BPMJ-12-2018-0380

Signavio Whitepaper. 2019. Putting the 'P' in RPA: overcoming the challenges of RPA Implementation. Technical report, Signavio GmbH. Saatavilla: https://www.signavio.com/downloads/white-papers/overcoming-challenges-rpa-implementation/

Silver, B. 2011. BPMN Methods & Style with BPMN Implementer's Guide. 2nd edition Cody-Cassidy Press, USA. ISBN: 978–0–9823681–1–4.

Singh, S.K., Rathore, S. & Park, J.H. 2019. BlockIoTIntelligence: A Blockchain-enabled Intelligent IoT Architecture with Artificial Intelligence. Future Gener. Comput. Syst. Vol. 110, 721–743. Saatavilla: https://doi.org/10.1016/j.future.2019.09.002

Suri, V.K., Elia, M. & van Hillegersberg, J. 2017. Software Bots - The Next Frontier for Shared Services and Functional Excellence. Industrial Engineering & Business Information Systems. Vol. 306, 81-94. Saatavilla: https://www.scopus.com/record/display.uri?eid=2-s2.0-85036628221&origin=inward&txGid=083cbdf5dbb408a70f63353342cb5b9a

Tilastokeskus. 2023. Luokitukset. [Työelämän tilastotietoa]. Saatavilla: https://www.stat.fi/til/tyti/2013/13/tyti_2013_13_2014-04-01_kat_007_fi.html

Tornbohm, C. 2017. Market guide for robotic process automation software. Study, Gartner Research. Saatavilla: https://www.gartner.com/en/documents/3835771

van der Aalst, W.M.P., Bichler, M. & Heinzl, A. 2018. Robotic process automation. Business & Information Systems Engineering. Vol. 60(4), 269–272. Saatavilla: https://doi.org/10.1007/s12599-018-0542-4

Wanner, J., Hofmann, A., Fischer, M., Imgrund, F., Janiesch, C. & Geyer-Klingeberg, J. 2019. Process selection in RPA projects-towards a quantifiable method of decision making. 40th International Conference on Information Systems (ICIS). Munich, Germany. Saatavilla: https://www.researchgate.net/publication/337289984_Process_Selection_in_RPA_Projects_-_Towards_a_Quantifiable_Method_of_Decision_Making

Wellman, C., Stierle, M., Dunzer, S. & Martzner, M. 2020. A Framework to Evaluate the Viability of Robotic Process Automation for Business Process Activities. Institute of Information Systems, Friedrich-Alexander-Universität. Asatiani, A., et al. (Eds.) 2020. BPM Blockchain and RPA Forum 2020, LNBIP Vol. 393, 200–214. Saatavilla: https://doi.org/10.1007/978-3-030-58779-6_14

Yatskiv, S., Voytyuk, I., Yatskiv, N., Kushnir, O., Trufanova, Y. & Panasyuk, V. 2019. Improved method of software automation testing based on the robotic process automation technology. IEEE. 9th International Conference on Advanced Computer Information Technologies (ACIT), pp. 293–296. Saatavilla: https://www.semanticscholar.org/paper/Improved-Method-of-Software-Automation-Testing-on-Yatskiv-Voytyuk/d41aa729e8bdbcb7a632092638a89f9bf0695f35

YTN. 2023. Ylempi toimihenkilö. [Ammattiliiton verkkosivu]. Saatavilla: https://ytn.fi/ukk/ylempi-toimihenkilo/

Zhu, L., Osterweil, L., Staples, M., Kannengiesser, U. & Simidchieva, B. 2007. Desiderata for Languages to be Used in the Definition of Reference Business Processes. Int. J. Software Informatics. Vol. 1(1), 37-65. Saatavilla: https://www.researchgate.net/publication/220130627_Desiderata_for_Languages_to_be_Used_in_the_Definition_of_Reference_Business_Processes

Liitteet

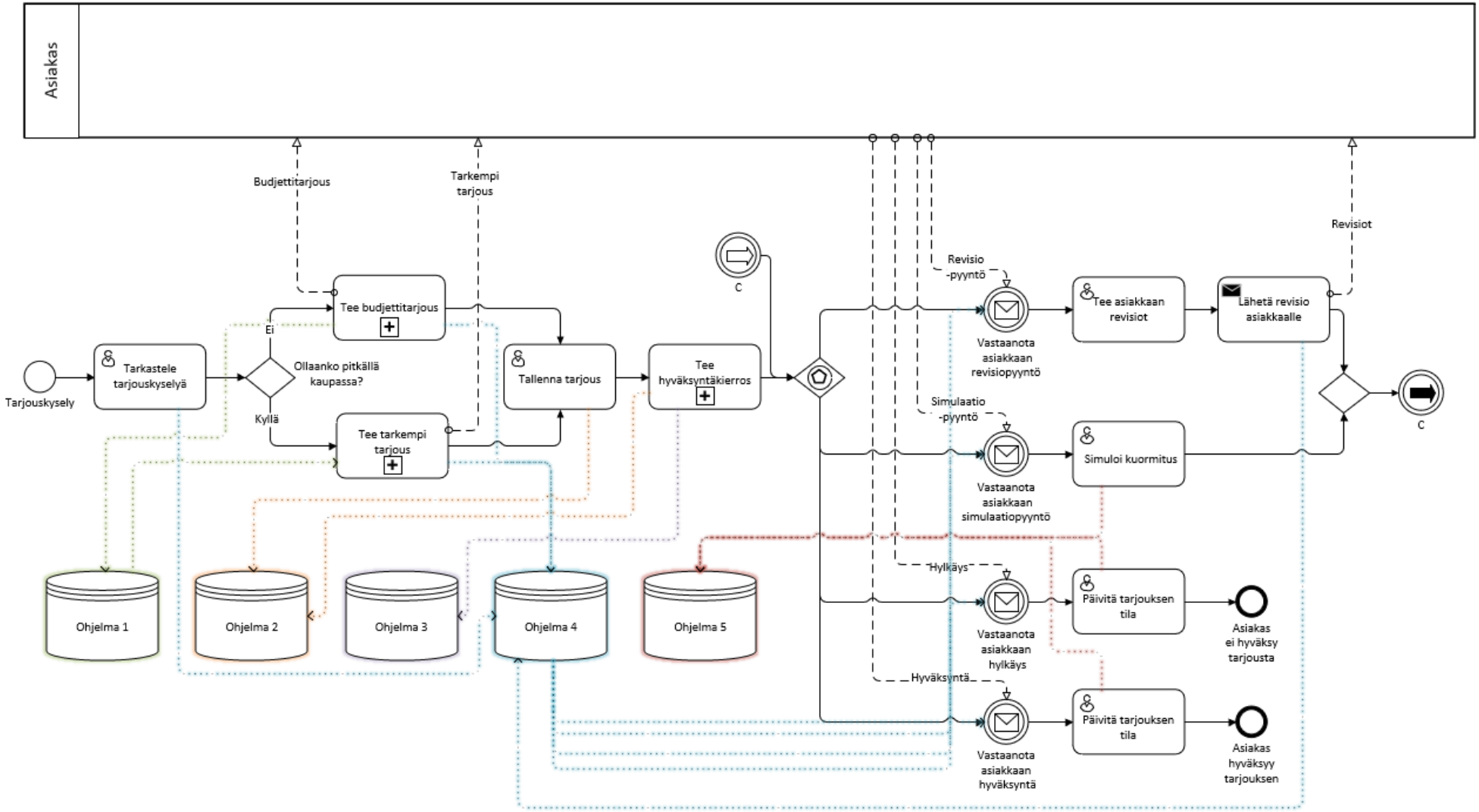
Liite 1. Prosessimallinnuksen haastattelukysymykset

Kun tarkastellaan osaprosessia,
• Mistä tiedetään, että prosessi on alkanut?   ○ Voiko prosessi alkaa useammalla eri tavalla?
• Mitä resursseja tarvitaan prosessin suorittamiseen?   ○ Inhimillisiä, aineellisia, taloudellisia, jne.
• Mitä tietoa tarvitaan prosessin suorittamiseen?   ○ Asiakkailta, toimittajilta, sisäisesti, jne.

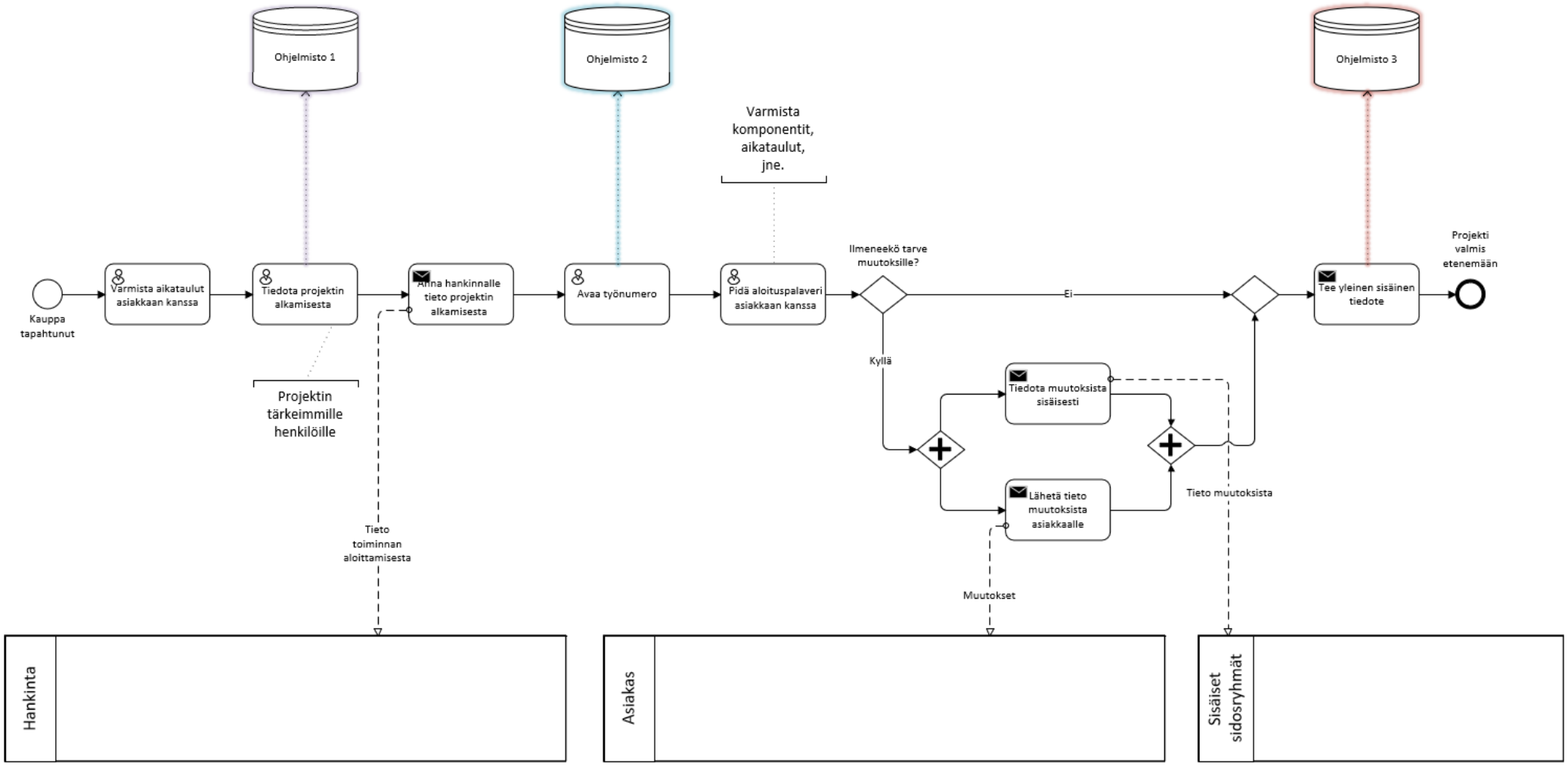
• Kuinka paljon kalenteri- ja työaikaa prosessin läpiviennissä suurin piirtein menee
  ○ ihanteellisessa tilanteessa?
  ○ epäihanteellisessa tilanteessa?
  ○ tavallisessa tilanteessa?

• Mihin osaan prosessia menee keskimäärin eniten aikaa?
• Mikä on tärkein osa prosessia?
• Mitä ohjelmistoja käytetään prosessin suorittamisessa?
• Mitkä ulko- tai sisäpuoliset asiat voivat aiheuttaa poikkeuksia prosessissa?
• Mistä tiedetään, että prosessi on päättynyt?   ○ Voiko prosessi päättyä useammalla eri tavalla?
• Mikä kysymys tästä listasta puuttui?   ○ Mistä olisi hyötyä myös tulevilla kysymyskierroksilla?

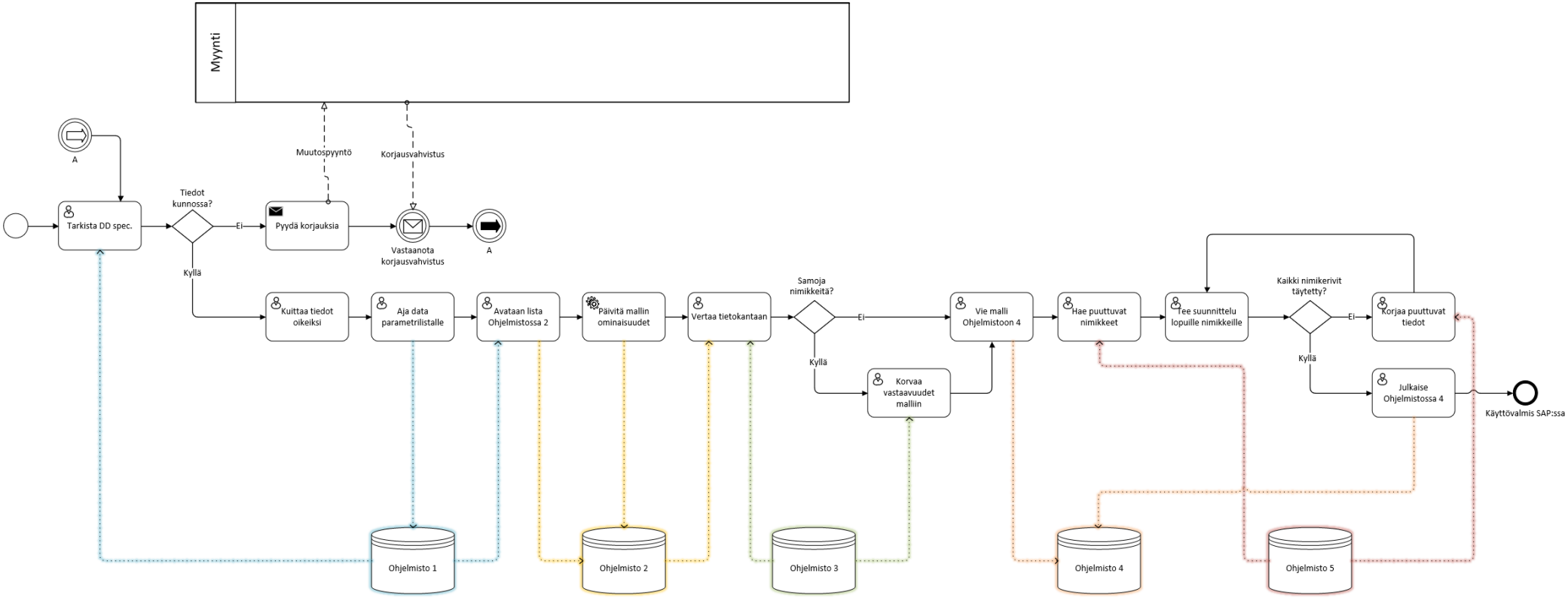
Liite 2. Tarjous-myyntiprosessi



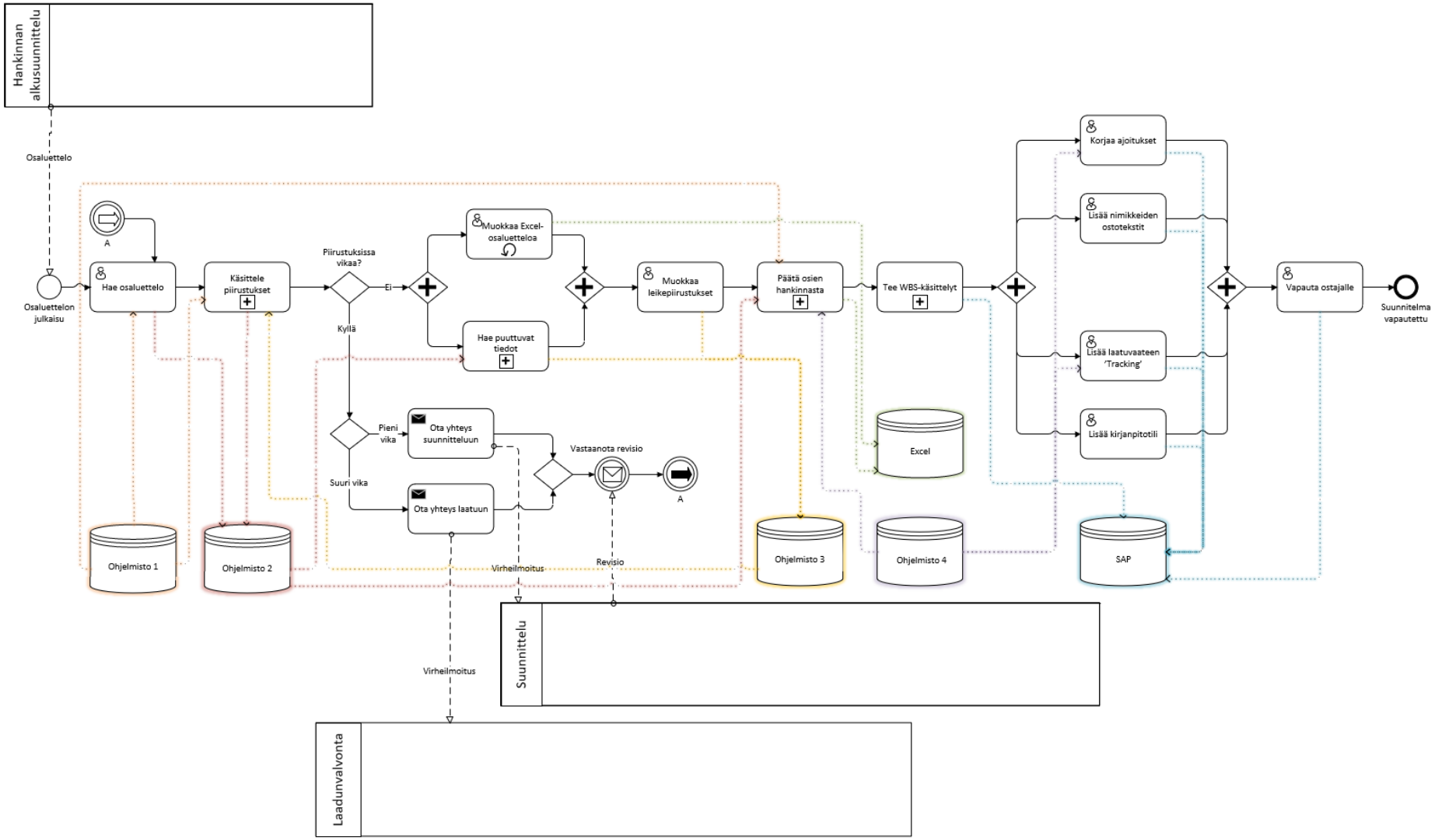
Liite 3. Projektin alkutoimenpiteet



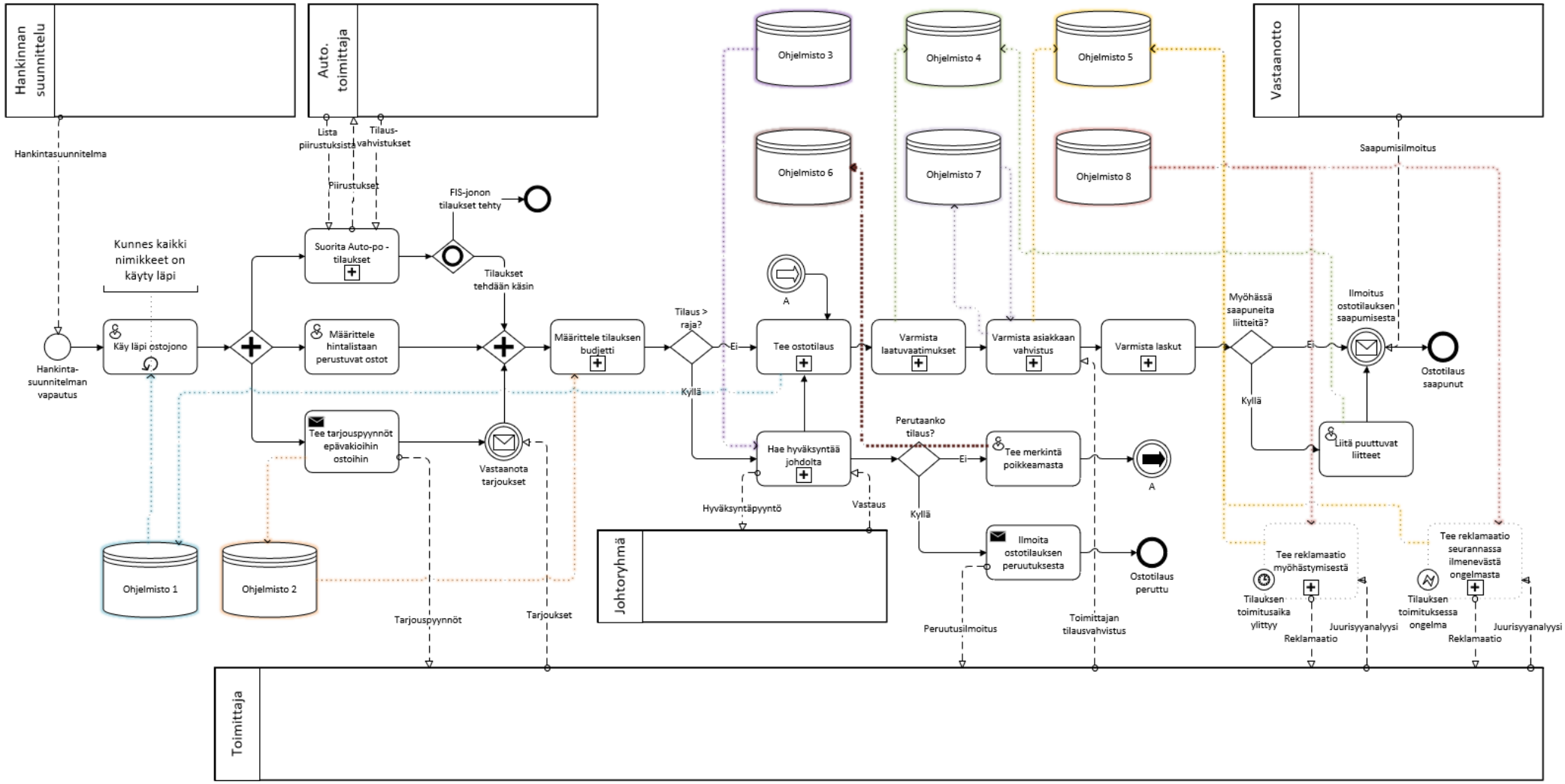
Liite 4. Hankinnan alkusuunnittelu



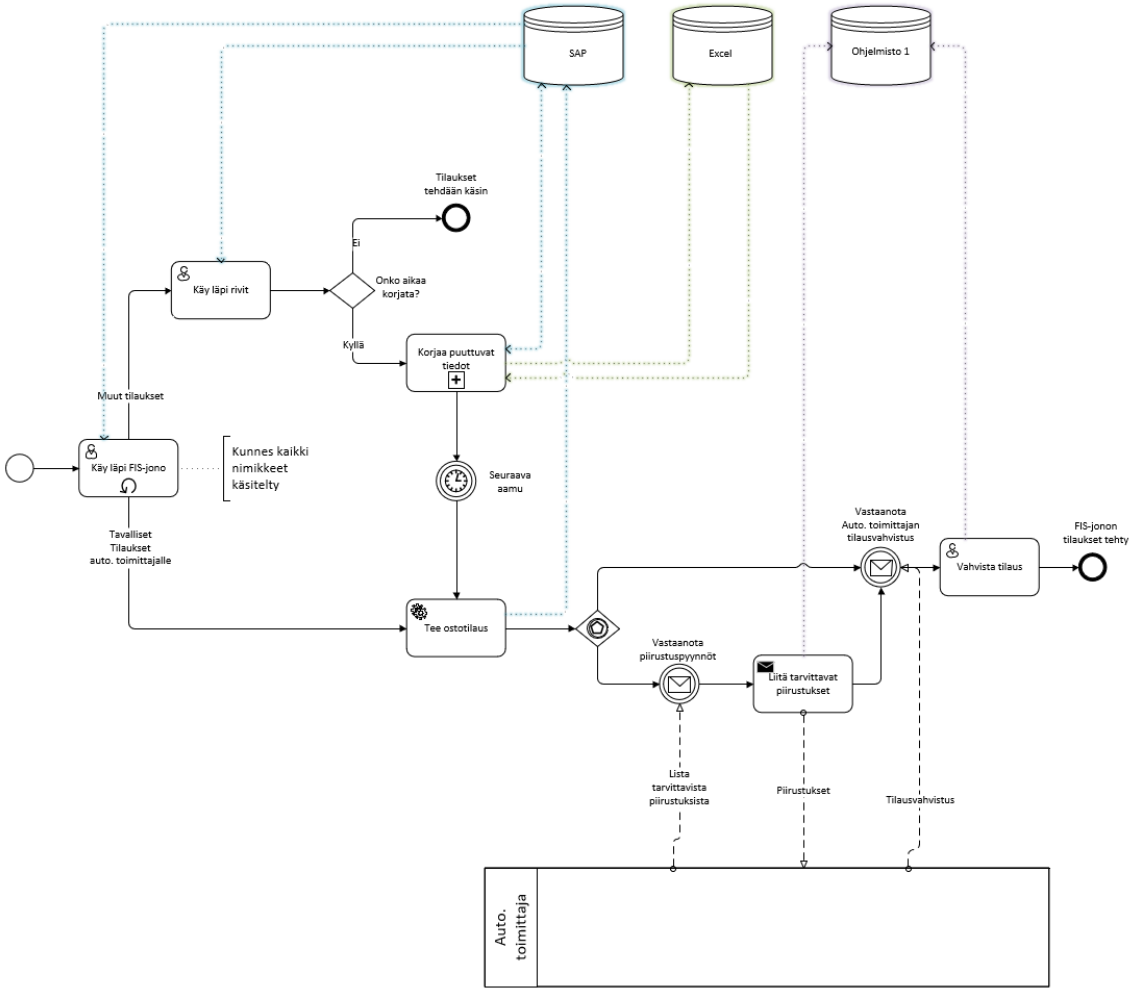
Liite 5. Hankinnan suunnittelu



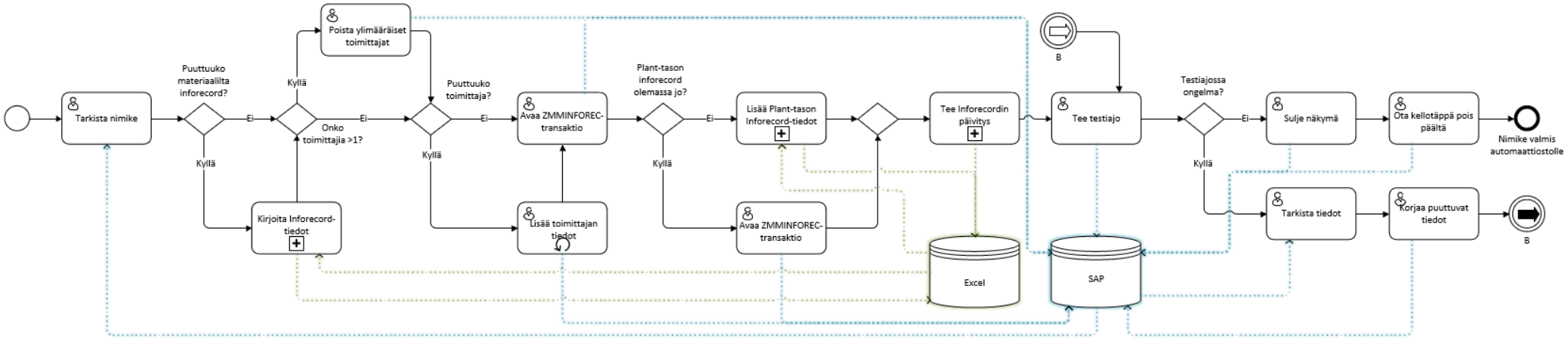
Liite 6. Operatiivinen osto



Liite 7. Tee auto-po-tilaukset



Liite 8. Korjaa puuttuvat tiedot



Liite 9. Pätkä Luo_Excelit.py ohjelmakoodista, josta luottamuksellinen tieto sekä käyttäjänimet on piilotettu

```python
#-*- coding: UTF-8 -*-
from __main__ import *
import openpyxl, time
alku = time.time()

#Haetaan ostojonotiedosto
polku = r'C:\[redacted]Ostojono.xlsx'
wb = openpyxl.load_workbook(polku)
sheet = wb.active

#ALIOHJELMAT

#Aliohjelma tietojen hakemiselle ostojonosta
def hae_tieto(sivu_luku, sarake_luku, sivu_kirjoitus, sarake_kirjoitus):
    arvot = []
    for column in sivu_luku.iter_cols():
        sheet_name = column[0].value
        if sheet_name == sarake_luku:
            for i, c in enumerate(column):
                if i == 0:
                    continue
                arvot.append(c.value)
            else:
                continue
    j = 1
    for column in sivu_kirjoitus.iter_cols():
        sheet_name = column[0].value
        if sheet_name == sarake_kirjoitus:
            for i in column:
                i = 2
                for arvo in arvot:
                    sivu_kirjoitus.cell(row = i, column = j).value = str(arvo)
                    i += 1
                else:
                    continue
        else:
            j += 1
            continue

#Aliohjelma vakiosarakkeiden täytölle
def tayta_sarake(sivu, sarake, arvo):
    for column in sivu.iter_cols():
        column_name = column[0].value
        if column_name == str(sarake):
            for i, cell in enumerate(column):
                if i == 0:
                    continue
                elif cell.value != None and cell.value != 'FIM' and cell.value != 'FIL':
                    continue
                cell.value = str(arvo)

#Aliohjelma sarakkeen lisäämiseen annetulle sivulle, annettuun kohtaan ja annetulla nimellä
def lisaa_sarake(sivu, indeksi, sarakkeen_nimi):
    sivu.insert_cols(indeksi)
    sivu.cell(row = 1, column = indeksi).value = sarakkeen_nimi
```

```python

#Aliohjelma sarakkeiden kopiointiin ja kirjoittamiseen
def kopioi_sarake(sivu, sarake_luku, sarake_kirjoitus):
    j = 1
    for column in sivu.iter_cols():
        column_name = column[0].value
        if column_name == sarake_kirjoitus:
            sarake = j
        else:
            j += 1
            continue
    for column in sivu.iter_cols():
        column_name = column[0].value
        if column_name == sarake_luku:
            for i, cell in enumerate(column):
                if i == 0:
                    continue
                sivu.cell(row = i + 1, column = sarake).value = str(cell.value)
            else:
                continue
        else:
            continue

```

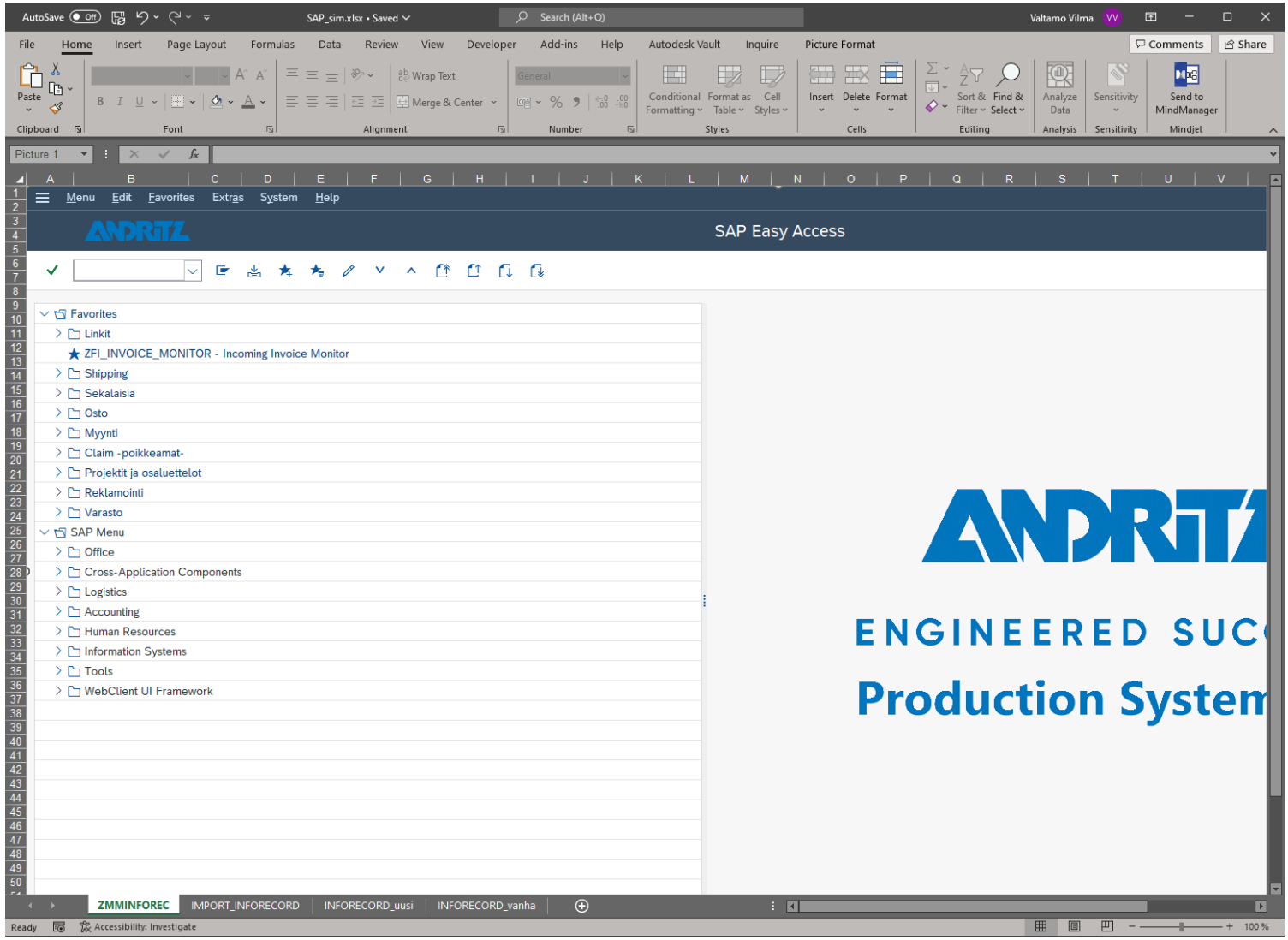
Liite 10. Pätkä Haku.py ohjelmakoodista, josta luottamuksellinen tieto sekä käyttäjänimet on piilotettu

```python
#-*- coding: UTF-8 -*-
wbs = '' #Työnumero
from __main__ import *
import openpyxl, time, pprint
alku = time.time()

#Avataan Hankintasuunnitelma
wb = openpyxl.load_workbook(r'C:\[redacted]\Hankintasuunnitelma.xlsx', data_only=True)

#Määritellään työkirjan sivut muuttujiksi ja lisätään ne listaan
sheet_1 = wb[[redacted]]
sheet_2 = wb[[redacted]
sheet_3 = wb[[redacted]
sheet_4 = wb[[redacted]
sheet_5 = wb[[redacted]

sivut = []
sivut.append(sheet_1)
sivut.append(sheet_2)
sivut.append(sheet_3)
sivut.append(sheet_4)
sivut.append(sheet_5)

#Aliohjelma arvon hakuun työkirjan eri sivuilta
def hae_sivulta(sivu):
    for row in range(2, sivu.max_row + 1):
        if sivu.cell(row = row, column = 2).value == wbs[0:15]:
            hae_sivulta.aloitus = row
            for r in range(hae_sivulta.aloitus + 1, sivu.max_row + 1):
                cell = sivu.cell(row = r, column = 2)
                if 'P-' in str(cell.value) or 'H-' in str(cell.value) or 'I-' in str(cell.value):
                    hae_sivulta.lopetus = r
                    break
                else:
                    continue
        else:
            continue

```

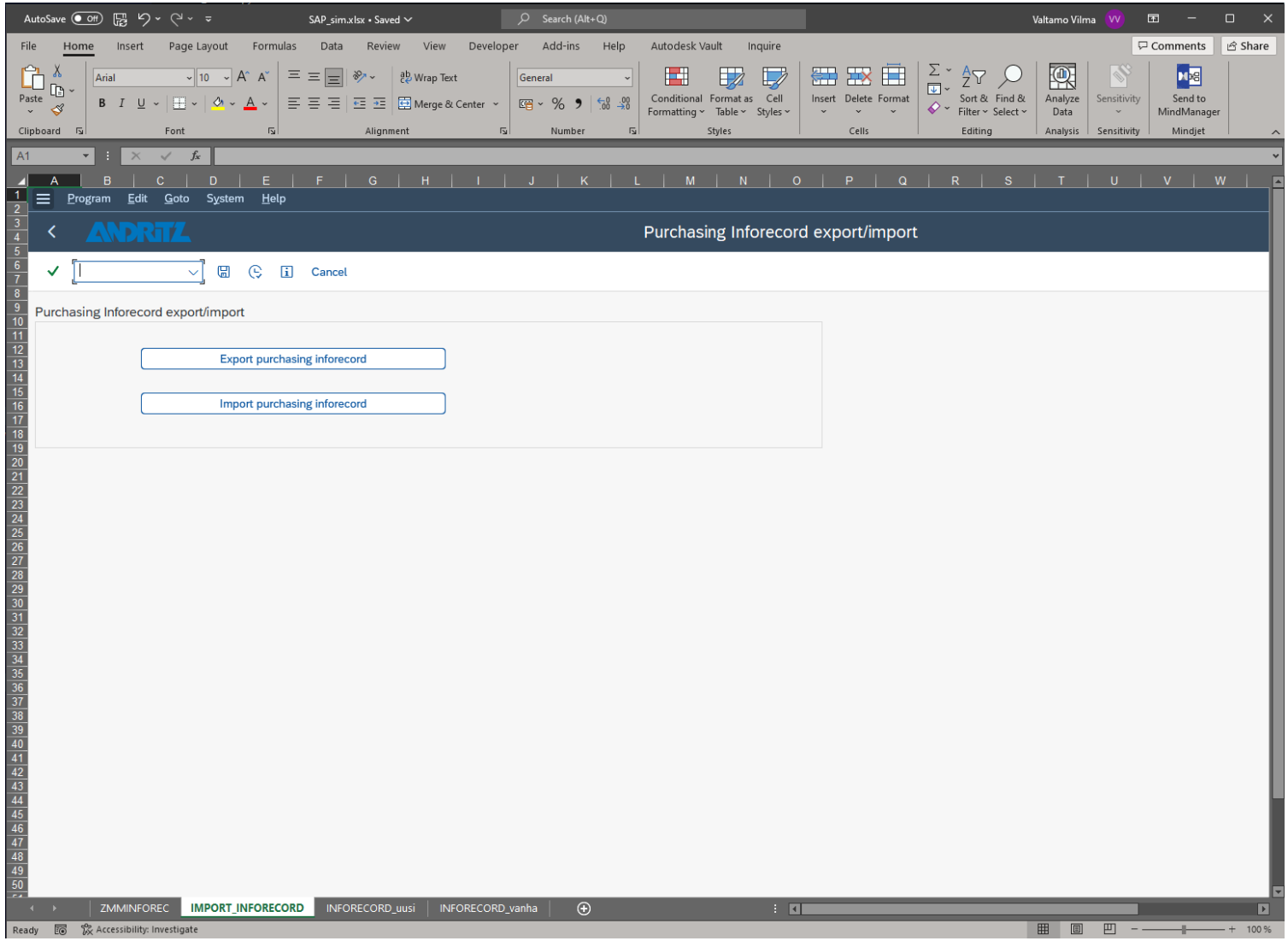
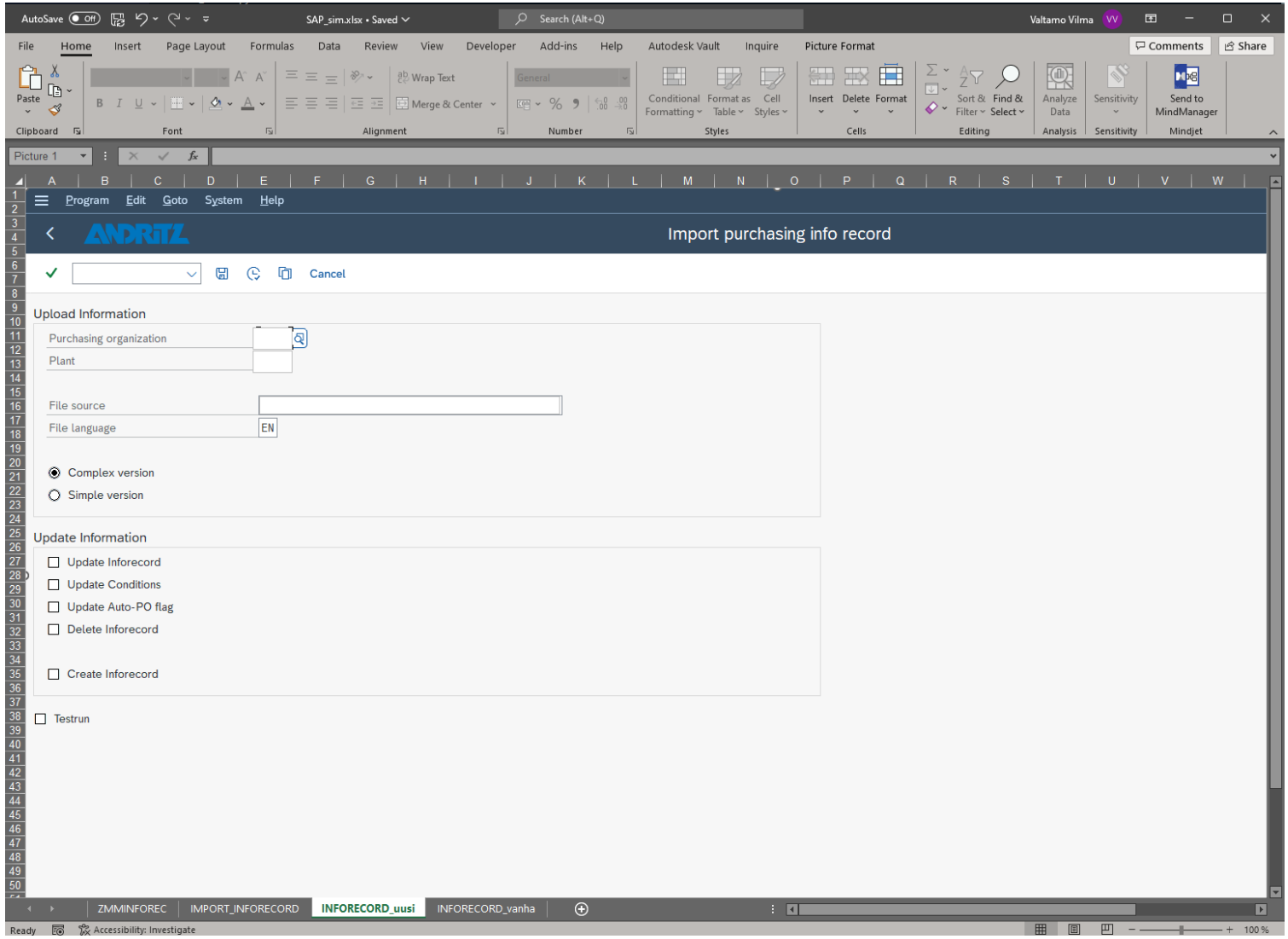
Liite 11. Pätkä SAP_robotti.py ohjelmakoodista, josta luottamuksellinen tieto sekä käyttäjänimet on piilotettu

```
#-*- coding: UTF-8 -*-
from __main__ import *
import pyautogui, time
alku = time.time()

#Ilmoitetaan käyttäjälle, miten robotti pysäytetään
pyautogui.FAILSAFE = True
print('Robotin pysäyttämiseksi, siirrä hiiri ruudun vasempaan ylänurkkaan.')

#Selvitetään käyttäjältä, kuinka monta tiedostoa syötetään
t = input('Kuinka monta tiedostoa syötetään? (1-3)\n')
if t == '0':
    print('Ohjelma lopetetaan.')
    exit()
while t !='1' and t != '2' and t != '3':
    print('Väärä syöttö, yritä uudelleen.')
    t = input('Kuinka monta tiedostoa syötetään?\n')

#Lähdetään SAP-etusivulta kirjautuneena

#Avataan ZMMINFOREC-transaktio
im = pyautogui.locateOnScreen('Check.PNG')
pyautogui.moveTo(im)
pyautogui.doubleClick(pyautogui.moveRel(50, 0, duration=0.25))
pyautogui.typewrite('ZMMINFOREC')
pyautogui.click(pyautogui.moveRel(-50, 0, duration=0.25))
time.sleep(0.5)

#Painetaan 'Import purchasing inforecord' -nappia
pyautogui.click(pyautogui.moveRel(150, 150, duration=0.25))
time.sleep(0.5)
pyautogui.moveTo('Kello_uusi.PNG')
#Kirjoitetaan purchasing organization
pyautogui.click(pyautogui.moveRel(10, 85, duration=0.25))
pyautogui.typewrite('[redacted])
#Kirjoitetaan plant
pyautogui.click(pyautogui.moveRel(0, 30, duration=0.25))
pyautogui.typewrite('[redacted])
#Painetaan 'Simple version' -ruutu aktiiviseksi
pyautogui.click(pyautogui.moveRel(-255, 155, duration=0.25))

```

Liite 12. Simuloitu SAP Production System -etusivu



Liite 13. Simuloitu ZMMINFOREC-transaktion sivu



Liite 14. Simuloitu Import purchasing Inforecord -sivu