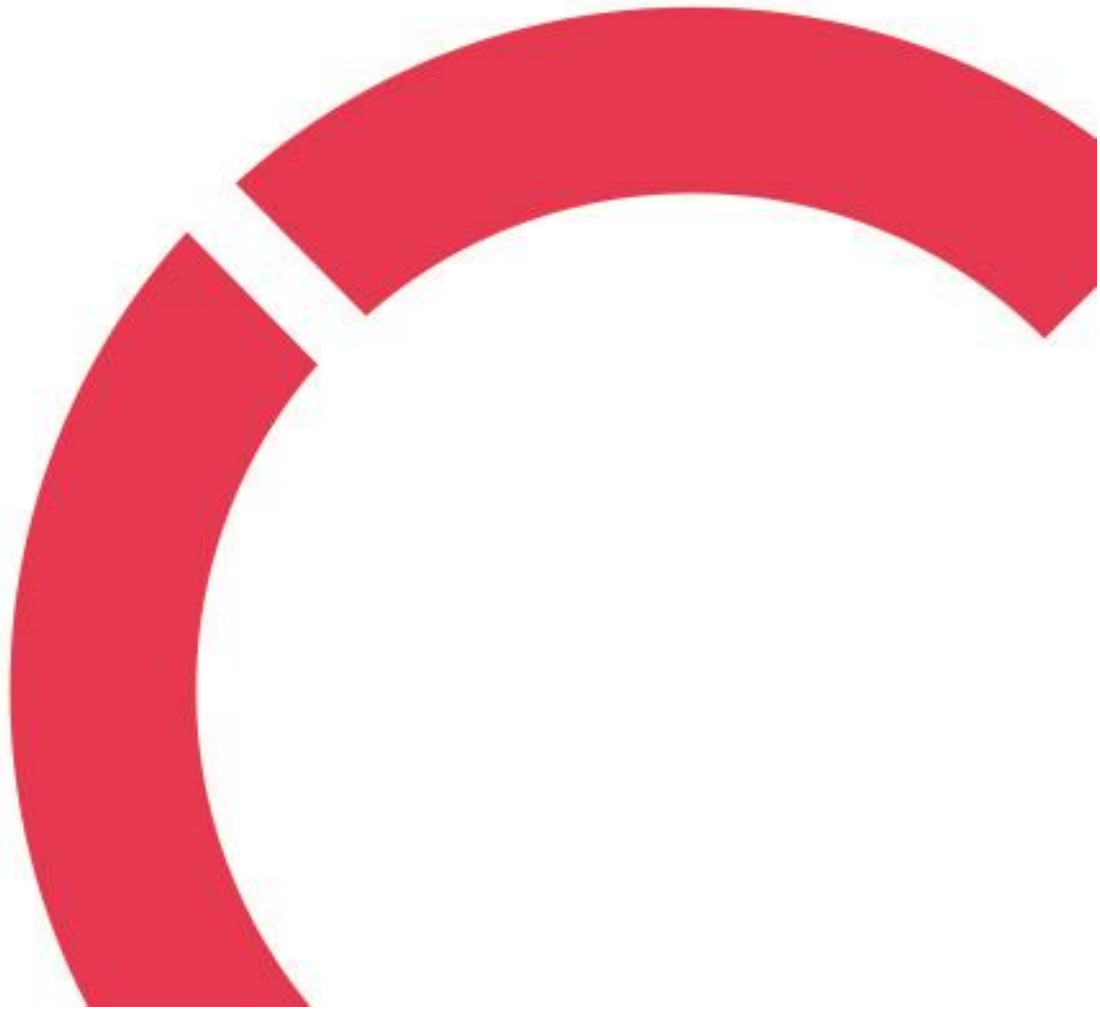


Antti Lidström

PROSESSILOUHINNAN MAHDOLLISTAMINEN ERP:SSÄ

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Digitalisaation johtaminen
Lokakuu 2024**



Centria-ammattikorkeakoulu	Aika Lokakuu 2024	Tekijä/tekijät Antti Lidström
Koulutus Digitalisaation johtaminen		<input type="checkbox"/> AMK <input checked="" type="checkbox"/> YAMK
Työn nimi PROSESSILOUHINNAN MAHDOLLISTAMINEN ERP:SSÄ		
Työn ohjaaja Pekka Makkonen		Sivumäärä 40
Työelämäohjaaja -		
<p>Tämä opinnäytetyö keskittyy prosessilouhinnan mahdollistamiseen ERP-järjestelmissä, erityisesti Microsoft Dynamics 365 Business Central -järjestelmässä. Prosessilouhinta tarjoaa tehokkaan menetelmän liiketoimintaprosessien analysointiin ja jatkuvaan parantamiseen hyödyntämällä eri tietojärjestelmistä luotuja tapahtumalokeja.</p> <p>Työssä keskityttiin yksinomaan Business Central -järjestelmään, eikä työssä esitetyt havainnot ja ratkaisut sovellu suoraan muihin ERP-järjestelmiin. Tutkimuksen odotettuina tuloksina ovat viitekehyksen kehittäminen ERP-tietojen poimimiseksi ja saavutettavaksi prosessilouhinnan työkaluja varten. Toteutettu ratkaisu tarjoaa käytännön näkemyksiä ja ohjeita organisaatioille, jotka haluavat hyödyntää ERP-järjestelmistä saatavaa dataa prosessilouhinnassa.</p>		
Asiasanat ERP-järjestelmät, liiketoimintaprosessit, prosessilouhinta		

ABSTRACT

Centria University of Applied Sciences	Date October 2024	Author Antti Lidström
Degree programme Digitalization management		
Name of thesis ENABLING PROCESS MINING IN ERP		
Centria supervisor Pekka Makkonen	Pages 40	
Instructor representing commissioning institution or company -		
<p>This thesis focuses on enabling process mining in ERP systems, specifically Microsoft Dynamics 365 Business Central. Process mining provides an efficient method for analyzing and continuously improving business processes by leveraging transaction logs generated from different information systems.</p> <p>The work focused exclusively on Business Central, and the findings and solutions presented in the thesis are not directly applicable to other ERP systems. The expected results of the study are the development of a framework for extracting and accessible ERP data for process mining tools. The implemented solution provides practical insights and guidance for organizations that want to utilize data from ERP systems in process mining.</p>		
Key words Business process development, ERP-systems, process mining		

KÄSITTEIDEN MÄÄRITTELY

ADX

Azure Data Explorer on Microsoft Azuren tarjoama data-analytiikkapalvelu, joka mahdollistaa suurten tietomäärien reaaliaikaisen analysoinnin. Se on suunniteltu erilaisten tietojoukkojen tehokkaaseen kyselyyn ja analysointiin.

AL

Application Language on Microsoft Dynamics 365 Business Centralin ohjelmointikieli, jota käytetään Business Centralin laajennuksien ja sovellusten kehittämiseen.

BPM

Business Process Management on systemaattinen lähestymistapa organisaation liiketoimintaprosessien parantamiseen.

BPMN

Business Process Model and Notation on liiketoimintaprosessien mallinnusstandardi, joka tarjoaa graafisen merkinnän liiketoimintaprosessien määrittämiseen. Open Management Group (<https://www.omg.org/>) on organisaatio, joka edustaa Business Process Modeling and Notation (BPMN) standardia.

IEEE

Institute of Electrical and Electronics Engineers IEEE vaalii, kehittää ja edistää globaalien teknologioiden rakentamista. IEEE SA on johtava alan standardien kehittäjä monenlaisissa teknologioissa

KQL

Kusto Query Language on tehokas ja intuitiivinen kyselykieli, jota käytetään suurten tietojoukkojen kyselyyn ensisijaisesti Microsoftin Azure Data Explorerissa ja muissa KQL:ää tukevista palveluissa, kuten Azure Monitor, Log Analytics sekä Application Insights, jota tässä työssä hyödynnetään.

XES

eXtensible Event Stream on suunniteltu tallentamaan ja hallitsemaan tapahtumalokitietoja, jotka ovat välttämättömiä prosessien louhintatoiminnoille.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
1.1 Tavoitteet	2
1.1.1 Tutkimusmenetelmä	3
1.1.2 Tiedonhankintaratkaisun toteuttaminen	3
1.1.3 Prosessianalyysi pullonkaulojen tunnistamiseksi	3
1.2 Rajaukset	4
1.2.1 Määritelty prosessit.....	4
1.3 Tietojen saatavuus ja laatu.....	4
1.3.1 Organisaatiokonteksti.....	4
1.3.2 Tietolähteet	5
1.3.3 Viitekehykset	5
2 PROSESSILOUHINTA	6
2.1 Tapahtumalokit	13
2.2 Prosessimallit	17
2.2.1 Prosessin visualisointi	18
2.3 Prosessilouhinnan hyödyntäminen ERP:in yhteydessä	21
3 KÄYTÄNNÖN TOTEUTUS.....	22
3.1 Datamallin suunnittelu	24
3.1.1 Datan tuottaminen telemetrian avulla	24
3.1.2 Datan tuottaminen räätälöidyn ratkaisukokonaisuuden avulla	27
3.1.3 Datan analysointi.....	32
4 YHTEENVETO TULOKSISTA	36
LÄHTEET	40

KUVAT

KUVA 1. Prosessilouhinta "siltana" datatieteen ja prosessitieteen välissä. ... **Error! Bookmark not defined.**

KUVA 2a+b. Esimerkki "meluisasta" sosiaalisesta verkostosta, jossa käsitellään työtovereiden välistä asiakirjojen käsittelyä. / Esimerkki samasta sosiaalisesta verkostosta suuren prosessikehityksen jälkeen. **Error! Bookmark not defined.**

KUVA 3. Prosessilouhinnan kolmen päätyypin sijoittaminen yritys ympäristöön. **Error! Bookmark not defined.**

KUVA 4. Prosessilouhinnan päätyypit **Error! Bookmark not defined.**

KUVA 5. Esimerkki tapahtumalokien hierarkiasta **Error! Bookmark not defined.**

KUVA 6. Prosessimalli **Error! Bookmark not defined.**

KUVA 7. Microsoft Process Mining -työkalu..... **Error! Bookmark not defined.**

KUVA 8. Process mining työkalu **Error! Bookmark not defined.**

KUVA 9. Prosessin visualisointi ja vaiheiden tulkinta pm4py:llä **Error! Bookmark not defined.**

KUVA 10. Esimerkki pm4py työkalulla tehdystä prosessin visualisoinnista .. **Error! Bookmark not defined.**

TAULUKOT

TAULUKKO 1. Esimerkki tapauslokin datasta **Error! Bookmark not defined.**

TAULUKKO 2. Työnkulun loki..... **Error! Bookmark not defined.**

1 JOHDANTO

ERP-järjestelmät sekä niihin liitetyt sovellukset hallinnoivat liiketoiminnan tärkeimpiä ydinprosesseja. Tietoja tallennetaan prosessin eri vaiheissa hyvinkin tarkasti ja tyypillisesti tätä dataa ei nykytilassa osata hyödyntää tehokkaasti organisaatioissa. Prosessilouhinta tarjoaa tähän tehokkaan menetelmän liiketoimintaprosessien analysointiin ja jatkuvaan parantamiseen hyödyntämällä eri tietojärjestelmistä luotuja tapahtumalokeja. Tämä opinnäytetyö keskittyy ratkaisun toteuttamiseen tietojen poimintaa varten Microsoft Dynamics 365 Business Central -järjestelmästä ja järjestelmän datan tehokkaaseen hyödyntämiseen prosessilouhinnan työkaluilla. Tämän tutkimuksen ensisijaisena tavoitteena on kehittää ja esitellä käytännönläheinen lähestymistapa ERP-järjestelmän hyödyntämisestä prosessilouhinnassa. Tutkimus alkaa katsauksella miten eri järjestelmissä on hyödynnetty prosessilouhintaa ja kuinka näitä voitaisiin soveltaa Microsoft Dynamics 365 Business Central -järjestelmässä. Nykytilassa Business Central ei tarjoa riittävän kattavaa dataa prosessilouhinnan hyödyntämiseksi.

Tutkimusmetodologia koostuu useammasta vaiheesta. Ensimmäisenä vaiheena on ERP Datat saavutettavuus eri prosessivaiheissa. Tämä vaihe sisältää ratkaisun suunnittelun ja kehittämisen tietojen poimimiseksi ERP-järjestelmästä. Harkinnanvaraisesti sovitut prosessikokonaisuudet kehitetään ja mahdollistetaan tietojen analysointi prosessilouhinnan työkaluilla. Tämä auttaa myös ymmärtämään ERP-järjestelmän tietokantarakennetta, sekä järjestelmäprosessien vaiheita kuinka prosessilouhintaa voidaan hyödyntää erilaisissa prosessikokonaisuuksissa.

Tämän jälkeen tulee tietojen esikäsittely ja muuntaminen. Kun ERP-järjestelmän data on saatuna eri vaiheista tallennettua, tulee se vielä saada muunnettua oikeaan formaattiin prosessilouhinnan vaatimuksia vastaavaksi dataksi. Tämä tarkoittaa käytännössä sitä, että ERP-järjestelmän data transformoidaan hyödynnettävissä olevaan muotoon, jotta sitä voidaan hyödyntää.

Kun data on saatu järjestelmästä poimittua, on aika siirtyä seuraavaksi prosessin analyysiin ja visualisointiin. Muunnettu data analysoidaan eri prosessinlouhintatekniikoilla määritetyistä prosesseista. Prosessilouhinnan tekniikoita sovelletaan organisaation prosessien analysointiin, mukaan lukien mahdollisesti uusien prosessivariaatioiden löytäminen. Tyypillisesti ERP-

järjestelmän prosessit ovat suunniteltu sovellettavaksi tietyn kaavan mukaisesti, mutta todellisuudessa prosessissa on usein eri variaatioita ja muuttujia suunniteltuun prosessiin nähden. Tässä vaiheessa prosessilouhinnan avulla pyritään tuomaan esiin eri prosessivariaatiot ja eri päätöksentekovaiheet prosessikokonaisuuden osalta. Tällä tavoin pyritään tunnistamaan prosessien mahdollisia pullonkauloja tai epätehokkaita prosesseja kokonaisuudesta. Tutkimuksen odotettuina tuloksina ovat viitekehyksen kehittäminen ERP-tietojen poimimiseksi ja saavutettavaksi prosessilouhinnan työkaluja varten. Toteutettu ratkaisu tarjoaa käytännön näkemyksiä ja ohjeita organisaatioille, jotka haluavat hyödyntää ERP-järjestelmistä saatavaa dataa prosessilouhinnassa.

1.1 Tavoitteet

Nykyisellään liiketoimintaympäristöjen tiedon määrä järjestelmissä kasvaa eksponentiaalisesti, jolloin organisaatioiden kyky hyödyntää tätä olemassa olevaa tietoa prosessiensa optimoinnissa on kriittinen kilpailuetu. Enterprise Resource Planning (ERP) eli toiminnanohjausjärjestelmät keräävät valtavan määrän dataa organisaation eri toiminnoista, mutta tämän datan tehokas hyödyntäminen prosessien parantamiseksi on edelleen haaste. Prosessilouhinta tarjoaa mahdollisuuden analysoida ja ymmärtää organisaation prosesseja syvällisesti käyttämällä ERP-järjestelmistä saatavaa dataa. Tämän tutkimuksen tavoitteena on selvittää, mitä tietoja ERP-datasta tarvitaan prosessilouhinnan mahdollistamiseksi ja kuinka prosessilouhinnan avulla voidaan tunnistaa prosessivariaatioita ja pullonkauloja, jotka hidastavat tai haittaavat organisaation tehokkuutta.

Keskeisimmät tutkimuksen tavoitteet ovat:

- ERP-Datan tunnistaminen prosessilouhintaan.

Tutkimuksessa määritetään ja luokitellaan, mitkä ERP-datan osa-alueet ovat kriittisiä prosessilouhinnan kannalta. Tavoitteena on ymmärtää, miten eri datatyyppejen (esim. transaktiot, tapahtumalokit, käyttäjätoiminnot) analysointi voi tuottaa syvällistä tietoa prosessien realistisesta toiminnasta.

- Prosessivariaatioiden ja pullonkaulojen tunnistaminen.

Pyritään kehittämään metodologia / luodaan ratkaisu, jonka avulla prosessilouhinnan tekniikoita voidaan hyödyntää prosessivariaatioiden ja pullonkaulojen tunnistamisessa ERP-

järjestelmissä kerätystä datasta. Tämän tavoitteen saavuttaminen mahdollistaa organisaatioiden kyvyn tunnistaa ja korjata prosessien mahdollisia heikkouksia, mikä parantaa toiminnan tehokkuutta ja vähentää kustannuksia.

1.1.1 Tutkimusmenetelmä

Tutkimuksen metodologiana sovellettiin kvalitatiivisen tutkimuksen ja toiminnallisen tutkimuksen yhdistelmää. Työssä toteutettiin perusteellinen taustatutkimus, jossa analysoitiin prosessilouhinnan datamalli sekä siihen liittyvät tarpeet ja vaatimukset. Tämän analyysin pohjalta kehitettiin viitekehys, jonka mukaisesti määriteltiin ratkaisukokonaisuuden keskeiset dataelementit. Näiden dataelementtien perusteella suunniteltiin ja toteutettiin kokonaisratkaisu, joka mahdollistaa syvällisen prosessilouhinnan toteuttamisen.

1.1.2 Tiedonhankintaratkaisun toteuttaminen

Tämän opinnäytetyön ensisijainen tavoite on suunnitella ja toteuttaa ratkaisu tietojen poimimiseen Microsoft Dynamics 365 Business Centralista, ERP-järjestelmästä. Tämä edellyttää systemaattisen lähestymistavan kehittämistä organisaation prosesseihin, tapahtumiin ja toimiin liittyvien olennaisten tietojen hakemiseksi. Toteutetun ratkaisun tulee kattaa poimittujen tietojen eheys, tarkkuus, turvallisuus, sekä toistettavuus, mikä luo pohjan myöhemmille prosessilouhintaanalyysille.

1.1.3 Prosessianalyysi pullonkaulojen tunnistamiseksi

Toinen keskeinen tavoite on hyödyntää prosessilouhintatekniikoita poimittujen tietojen analysoimiseksi ja mahdollisten pullonkaulojen tai tehottomuuden tunnistamiseksi organisaation työnkuluissa. Visualisoimalla prosessivirtoja, poikkeamia ja suorituskykymittareita pyrimme saamaan käsityksen prosessien viivästysten tai puutteiden perimmäisistä syistä. Tämän analyysin avulla sidosryhmät voivat tehdä tietoon perustuvia päätöksiä ja toteuttaa kohdennettuja parannuksia prosessien tehokkuuden ja vaikuttavuuden parantamiseksi.

1.2 Rajaukset

Tämän opinnäytetyön merkittävin raja on keskittyminen yksinomaan Microsoft Dynamics 365 Business Centraliin ERP-järjestelmänä. Vaikka tämä järjestelmä on laajalti käytetty ja edustaa monien organisaatioiden ERP-ympäristöjä, tässä opinnäytetyössä esitetyt havainnot ja ratkaisut eivät välttämättä sovellu suoraan muihin ERP-järjestelmiin. Siksi tutkimuksen laajuus rajoittuu Business Centralin ominaisuuksiin ja toimintoihin.

1.2.1 Määritellyt prosessit

Toteutettavuuden ja hallittavuuden varmistamiseksi tässä opinnäytetyössä keskitytään ennalta määriteltyyn organisaation prosesseihin. Nämä prosessit valitaan huolellisesti niiden merkityksen, monimutkaisuuden ja mahdollisen organisaation suorituskykyyn kohdistuvan vaikutuksen perusteella. Tämä rajallinen soveltamisala tarkoittaa kuitenkin sitä, että tietyt organisaation prosesseja tai näkökohtia ei välttämättä käsitellä tässä opinnäytetyössä. On olennaista tunnustaa nämä rajoitukset ja ymmärtää, että havainnot eivät välttämättä kata organisaation prosessien täydellistä kokonaisuutta.

1.3 Tietojen saatavuus ja laatu

Toteutetun ratkaisun ja myöhempien prosessilouhinta-analyysien tehokkuus riippuu ERP-järjestelmän tietojen saatavuudesta ja laadusta. Tietojen täydellisyyteen, tarkkuuteen ja tarkkuuteen liittyvät rajoitukset voivat vaikuttaa havaintojen paikkansapitävyyteen ja luotettavuuteen. Tietojen laatuun liittyviä kysymyksiä pyritään käsittelemään esikäsittely- ja validointivaiheiden avulla. On kuitenkin tärkeää tunnustaa, että tiedon rajoitukset voivat vaikuttaa tutkimuksen tuloksiin.

1.3.1 Organisaatiokonteksti

Lopuksi tässä opinnäytetyössä esitetyt havainnot ja suositukset sijoittuvat tarkasteltavan organisaation kontekstiin. Vaikka prosessilouhinta-analyyseistä saaduilla oivalluksilla voi olla laajempia vaikutuksia, niitä tulisi tulkita ja soveltaa organisaation omassa kontekstissa. Tästä syystä tulosten yleistettävyyden muihin organisaatioihin saattaa olla rajoitettua, ja lisävalidointi voi olla tarpeen eri organisaatioympäristöissä.

1.3.2 Tietolähteet

Opinnäytetyön kokonaisuus koostuu useammasta eri kokonaisuudesta, joten tietolähteinä on käytettävä myös ERP-järjestelmän olemassa olevaa dokumentaatiota, sekä kyseiseen järjestelmään viittaavia kirjoja. Tieteelliset artikkelit, tutkimukset sekä ammattikirjallisuus auttavat prosessilouhinnan analysoimisessa, sekä ratkaisukokonaisuuden suunnittelussa. Tapaustutkimukset ja yritysesimerkit auttavat ymmärtämään miten liiketoiminnassa nykyään hyödynnetään prosessilouhintaa, sekä mitkä ovat oleellisimmat asiat mitä prosessilouhinnan työkaluilla pyritään selvittämään. Tämän pohjalta suunnittelen arkkitehtuurin, sekä määritän prosessit. Tässä asiassa turvaudun omaan ammattitaitoon, jonka olen saanut järjestelmän konsultoinnin puitteissa useamman vuoden ajalta.

1.3.3 Viitekehukset

Teoreettisena viitekehysenä hyödynnetään yleisiä parhaita käytäntöjä. Business Process Management (BPM) on laaja-alainen viitekehys, joka keskittyy organisaatioiden prosessien tehokkuuden parantamiseen. Tämän avulla pyritään tunnistamaan olemassa olevia prosessivirtoja. Prosessilouhinnan kirjallisuuden avulla pyritään saamaan tuotetusta datasta mahdollisimman kattava analyysi, miten prosessivirrat menevät liiketoiminnassa ja minkä tyypisiä pullonkauloja sieltä olisi tunnistettavissa. Toiminnallisessa viitekehyksessä kehitetään ratkaisu, joka tuottaa prosessivirtojen vaiheista tarvittavan datan, joka myöhemmin käsitellään prosessilouhinnan työkaluja käyttäen. Järjestelmäteknisesti hyödynnetään Microsoftin parhaita käytäntöjä, joista on kattava dokumentaatio Microsoftin omilla sivustoilla.

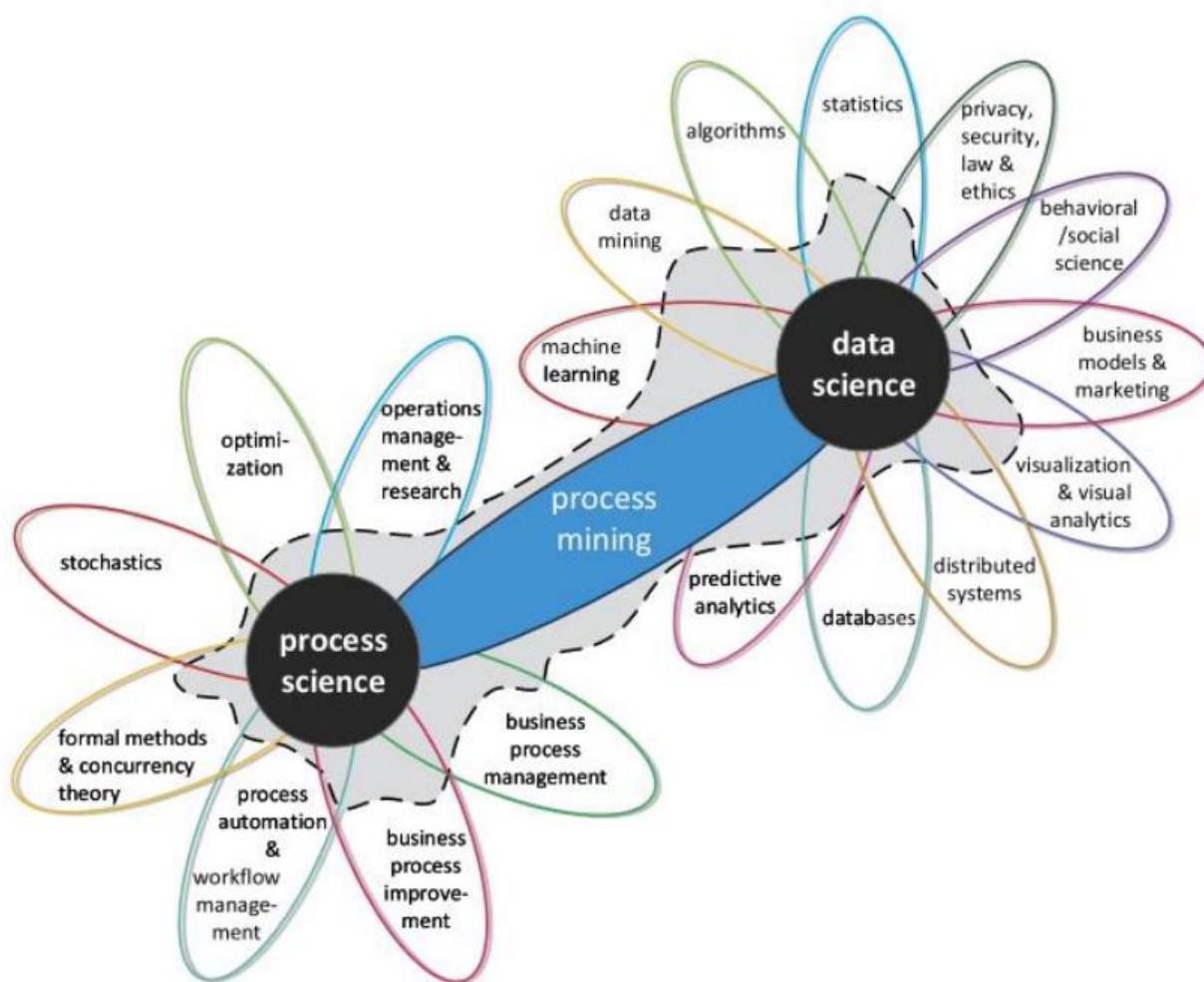
2 PROSESSILOUHINTA

Prosessilouhinta on noussut tehokkaaksi työkaluksi organisaatioille, jotka pyrkivät optimoimaan toimintaansa. Analysoimalla liiketoimintaprosessien jättämiä digitaalisia jalanjälkiä yritykset voivat tunnistaa tehottomuuksia, varmistaa vaatimuksenmukaisuuden ja parantaa asiakastyytyväisyyttä. Prosessilouhinnan tehokkuus kuitenkin edellyttää huolellista käyttötapausten valintaa.

Prosessilouhinta on suhteellisen uusi tieteenala. Sen tutkimus alkoi kehittyä merkittävästi vasta 2000-luvun lopulla, erityisesti sen jälkeen, kun sähkö- ja elektroniikkainstituutti (IEEE) perustettiin vuonna 2009 ja prosessilouhinnan manifesti julkaistiin vuonna 2011. Manifestin julkaisun myötä prosessilouhinnan keskeiset haasteet alkoivat saada enemmän huomiota. Aiheeseen liittyvä ensimmäinen merkittävä teos "Process Mining: Data Science in Action", julkaistiin vuonna 2016 Will van der Aalstin toimesta, ja siitä on tullut keskeinen viite prosessilouhinnan tutkimuksessa. Prosessilouhinnan manifesti määrittelee tieteenalan, joka sijoittuu prosessimallinnuksen, analyysin ja datatieteen, kuten laskennallisen älykkyyden ja analytiikan, leikkauspisteeseen.

Prosessilouhinnan perusta on digitaalisiin tapahtumalokeihin tukeutuminen. Nämä lokit ovat olennaisia, koska ne tallentavat prosessin sisällä tapahtuvien toimintojen sarjat. Prosessilouhintaan soveltuvat prosessit tulee valita sen perusteella, että tarvittava data on saatavilla prosessilouhinnan työkalujen käyttöön. Tämä tarkoittaa, että prosessin on tuotettava tarvittava data lokitietoineen. Lokien laatu, tarkkuus ja täydellisyys ovat keskeisessä asemassa. Lokien tulee tallentaa paitsi toimintojen järjestys myös kontekstuaaliset tiedot, kuten aikaleimat, osallistujat ja tulokset. Tämä tietojen rikkaus mahdollistaa prosessivirtojen rekonstruoinnin sekä poikkeamien tunnistamisen oletetuista tai esimääritetyistä prosessivirroista. (Van der Aalst 2016).

Van der Aalst (2011) huomauttaa, että prosessilouhinta voidaan nähdä "puuttuva linkki datatieteen ja prosessitieteen välillä". Tämä tarkoittaa, että prosessilouhinta yhdistää esimerkiksi datatieteen algoritmit ja tilastot prosessitieteen työnkulun hallintaan ja liiketoimintaprosessien hallintaan. Kuva 1 havainnollistaa prosessilouhinnan asemointia prosessitieteen ja datatieteen välillä.



KUVA 1. Prosessilouhinta "siltana" datatieteen ja prosessitieteen välissä (Van Der Aalst, 2016).

Koska prosessilouhinta yhdistää sekä datatieteen että prosessitieteen, sen keskeinen fokus sisältää laajasti muutakin kuin pelkästään prosessien tunnistamisen ja vaatimustenmukaisuuden. Tästä syystä voidaan sanoa, että prosessilouhinnan tavoitteet menevät päällekkäin datatieteen ja prosessitieteen lähestymistapojen, metodologioiden, periaatteiden, menetelmien, työkalujen ja paradigmojen kanssa.

Prosessilouhinta algoritmipohjaisena lähestymistapana mahdollistaa liiketoimintaprosessien mallintamisen tarkasti ja tehokkaasti. Prosessien mallintaminen, esimerkiksi ERP-järjestelmän kehittämisen yhteydessä, on usein monivaiheinen ja vaativa tehtävä, joka vaatii merkittävää ajallista ja taloudellista panostusta (Kapulin, Russikh ja Moor 2019). Van der Aalstin (2011) mukaan manuaalisesti laaditut prosessimallit eivät useimmissa tapauksissa

vastaa todellisuutta, vaan ne esittävät idealisoidun näkemyksen prosessista. Tämä idealisointi voi johtaa päätöksentekijöitä harhaan, mikäli parannukset kohdistuvat prosessin väärin osiin, mikä saattaa johtaa prosessin etääntymiseen sen todellisesta, optimaalisesta tilasta.

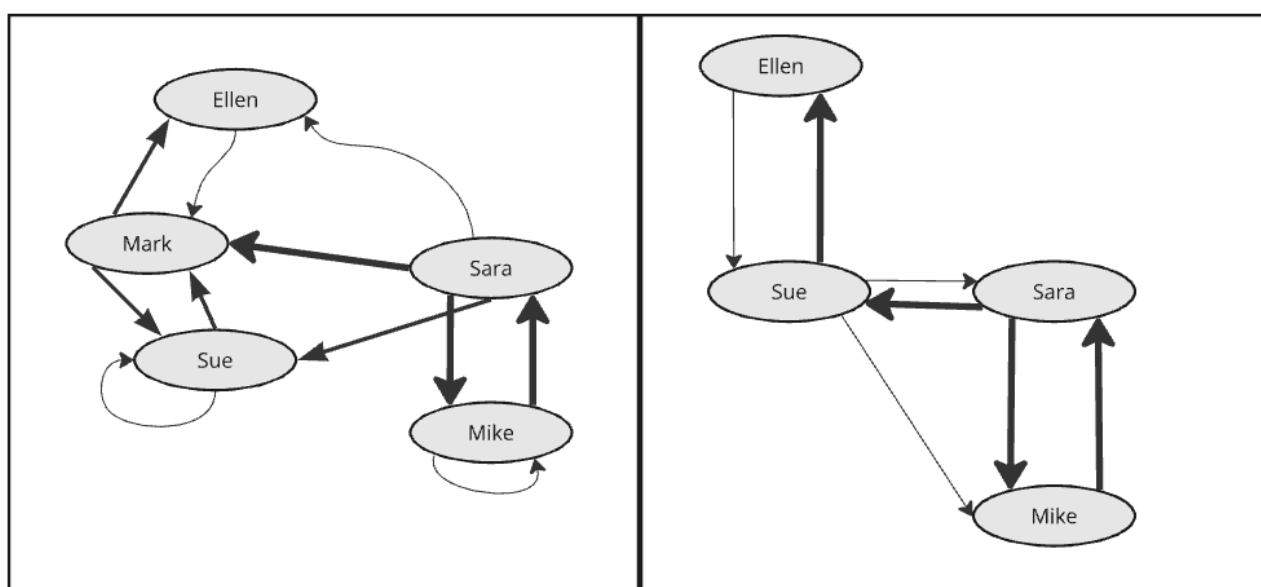
Prosessimallinnuksen korkean kustannustason ja manuaalisesti tuotettujen mallien rajoitteiden vuoksi on perusteltua hyödyntää tapahtumalokitietoja yhdessä prosessilouhinnan kanssa. Prosessilouhinta tarjoaa ratkaisun edellä mainittuihin ongelmiin, sillä sen ensisijaisena tavoitteena on poimia tapahtumalokien perusteella tietoa prosessien toiminnasta (Van der Aalst 2011). Esimerkiksi prosessilouhinnan työkalut voivat automaattisesti tuottaa prosessimalleja hyödyntämällä tapahtumalokien tietoja, joita kerätään esimerkiksi ERP- tai CRM-järjestelmistä. Tämä lähestymistapa nopeuttaa mallinnusprosessia ja tekee siitä vähemmän riippuvaisen inhimillisestä työpanoksesta. Jans ym. (2011) korostavat, että prosessilouhinnan objektiivisuus varmistaa mallien luotettavuuden, sillä mallit perustuvat puhtaasti dataan ilman tutkijan ennakko-oletuksia. Näin ollen prosessilouhinta mahdollistaa realististen prosessimallien muodostamisen, jotka eivät ole alttiita idealisoiduille käsityksille prosessien toiminnasta.

Günther ja Van der Aalst (2007) korostavat tutkimuksessaan, että prosessilouhinta on tutkimussuuntaus, joka pyrkii hyödyntämään lokitietoja prosessien abstraktien ja kompaktien esitystapojen poimimiseksi. He painottavat, että lokitiedot tulee ymmärtää historiallisina suoritus-tietoina, mikä antaa arvokasta tietoa menneistä tapahtumista. Wu (2007) puolestaan huomauttaa, että prosessilouhintaan perustuva liiketoimintaprosessianalyysi (BPA) keskittyy tapahtumahistoriaan, eikä näin ollen voi taata tulevien tapahtumien ennustettavuutta. Vaikka prosessilouhinta keskittyy historiallisiin tietoihin, jotka ovat helposti saatavilla yritysjärjestelmissä, se tarjoaa merkittäviä mahdollisuuksia organisaation sisäisen valvonnan ja prosessimuutosten seurannan kehittämiseen.

Jans ym. (2011) ja Van der Aalst (2011) tuovat esille, että prosessit, jotka noudattavat niin sanottua neljän silmän periaatetta, eli vaatimusta, että tietty asiakirja on hyväksyttävä vähintään kahden henkilön toimesta ennen lopullista hyväksyntää, voidaan tehokkaasti seurata sosiaalisen verkoston analyysin avulla. Kuvassa 2a esitetty sosiaalisen verkoston analyysi havainnollistaa neljän silmän periaatteen toteutumista. Jos tarkastellaan tilannetta, jossa sosiaalinen verkosto on muodostettu pelkästään tapahtumalokitietojen perusteella tietystä dokumenttivirrasta, ja prosessi on suunniteltu siten, että asiakirja pitäisi ensin tarkistaa Sue ja sen jälkeen Ellen, voidaan huomata, että suoraa yhteyttä (nuolta) Suen ja Ellenin välillä ei

ole. Tämä ei tarkoita, että Sue ja Ellen eivät tarkista asiakirjaa, mutta se osoittaa, että prosessi ei etene suunnitelmien mukaisesti ja vaatii tarkempaa tarkastelua.

Prosessilouhinnan avulla voidaan myös seurata ja analysoida prosessimuutosten vaikutuksia vertaamalla niitä alkuperäiseen prosessiin ennen muutoksia. Tällainen seurantamenetelmä auttaa päätöksentekijöitä arvioimaan, ovatko prosessiin tehdyt muutokset olleet positiivisia vai kielteisiä. Kuva 2b havainnollistaa kuvitteellisen prosessin kehitystä kuvan 2a esittämän alkuperäisen prosessin jälkeen, missä voidaan nähdä, että suurin asiakirjavirta kulkee nyt Suesta Elleniin, osoittaen, että prosessimuutoksilla on ollut vaikutusta dokumenttivirtaukseen.

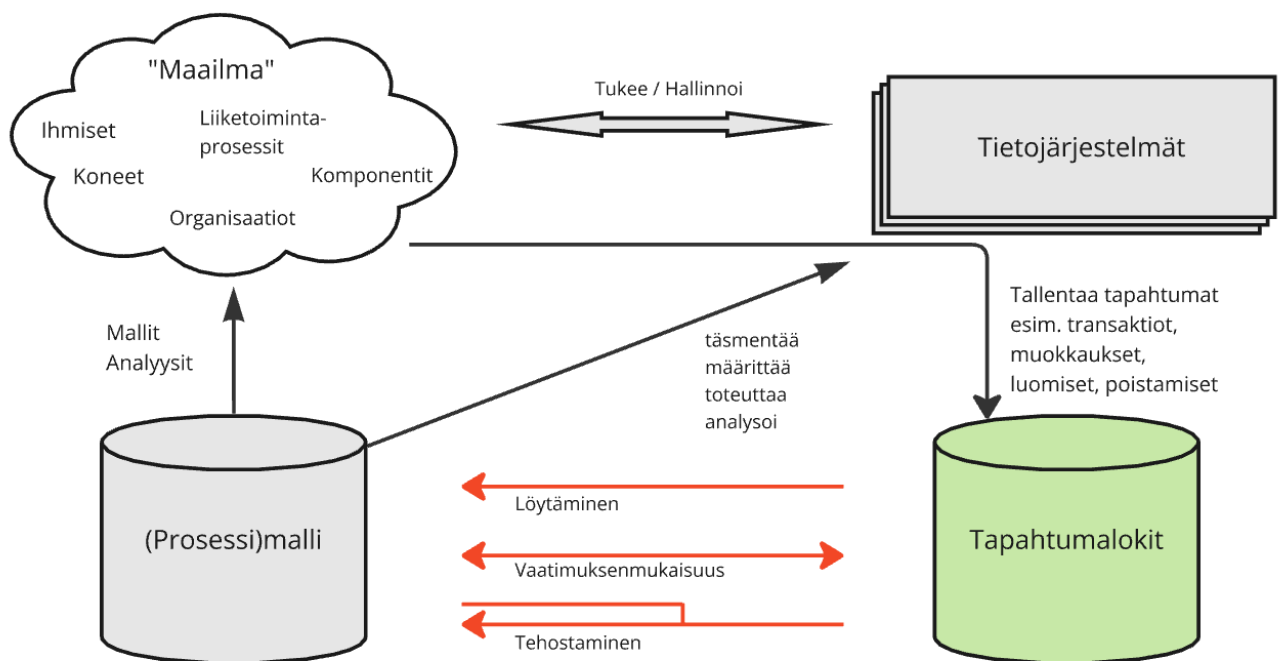


KUVA 2a+b. Esimerkki "meluisasta" sosiaalisesta verkostosta, jossa käsitellään työtovereiden välistä asiakirjojen käsittelyä. / Esimerkki samasta sosiaalisesta verkostosta suuren prosessikehityksen jälkeen.

Kuten aiemmin mainittiin, prosessilouhinta voi automaattisesti löytää prosessimalleja tapahtumalokitiedoista ilman ennakkotietoja. Tätä prosessilouhinnan tyyppiä kutsutaan prosessihakuksi (discovery), ja se on yksi kolmesta prosessilouhinnan perustyyppistä. Esimerkiksi Alpha-algoritmi, joka on yksi useista prosessilouhinnan algoritmeista, mahdollistaa tapahtumalokitietojen mallintamisen ja niistä johdetun prosessin esittämisen Petri-verkon muodossa. Tämän tyyppisen prosessilouhinnan avulla voidaan myös tunnistaa muita prosessimalleja, kuten sosiaalisia verkostoja.

On kuitenkin huomattava, että edistyneempien prosessimallien tunnistaminen vaatii usein kehittyneempiä tapahtumalokitetietoja, kuten henkilöstöresurssitietoja. Tapahtumalokin yksityiskohtaisuus vaikuttaa merkittävästi siihen, millaisia prosessimalleja voidaan tunnistaa. Discovery, eli etsintä, on yksi kolmesta prosessilouhinnan perustyyppistä (Van der Aalst 2011).

Kuva 3 havainnollistaa, kuinka kaikki kolme prosessilouhinnan perustyyppiä sijoittuvat (prosessi)mallin ja tapahtumalokin väliin. Kuva osoittaa, että prosessilouhinnan etsintätyyppiä käytetään ainoastaan prosessimallien luomiseen tapahtumalokitetiedoista, minkä vuoksi etsintää kuvaava nuoli suuntautuu tapahtumalokeista prosessimalleihin. Syöttö- ja lähtövirrat kuvassa 4 visualisoivat prosessivirran kulun kaikkien kolmen prosessilouhinnan tyyppin läpi ja osoittavat, että discovery-tyyppi vaatii vain tapahtumalokin tuottaakseen todellisen prosessimallin.



KUVA 3. Prosessilouhinnan kolmen päätyypin sijoittaminen yritys ympäristöön (Van Der Aalst 2011.)

Prosessilouhinnan toinen keskeinen tyyppi on vaatimustenmukaisuustarkistus (conformance checking), jonka avulla voidaan verrata tapahtumalokidatasta louhittua todellista prosessia olemassa olevaan ideaaliseen prosessimalliin (Sonawane ja Patki 2015; Van der Aalst 2011). Vaatimustenmukaisuustarkistus mahdollistaa yrityksille näkemyksen siitä, kuinka hyvin todel-

liset prosessit noudattavat määriteltyjä menettelytapoja, vai esiintyykö prosessien kiertämistä tai poikkeamia (Jans ym. 2011).

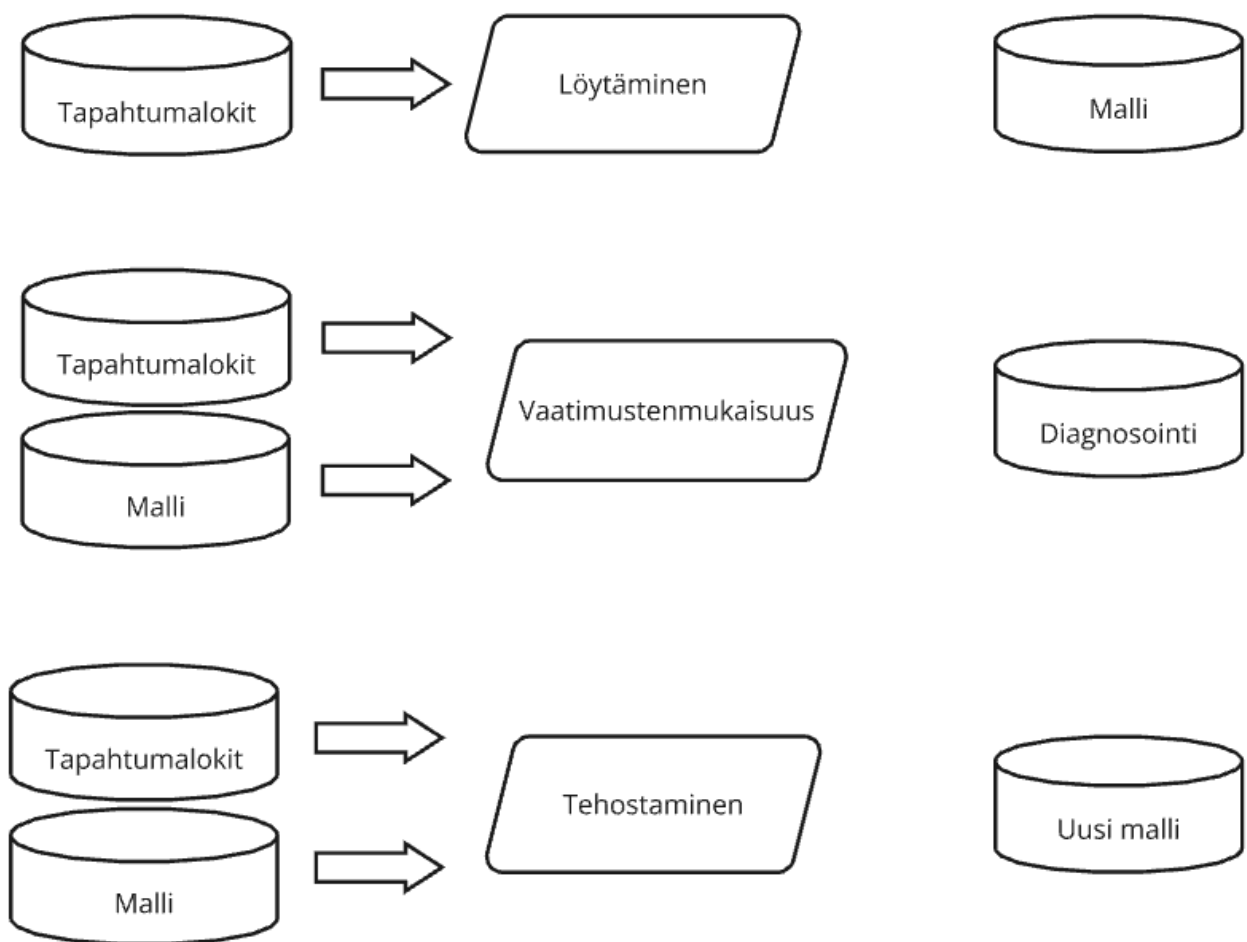
Kuva 3 havainnollistaa tätä prosessimallien ja tapahtumalokien välistä vertailua kaksisuuntaisena nuolena, joka kuvaa prosessi- ja tapahtumalokitetietojen välistä vuorovaikutusta. Esimerkiksi edellä mainittu neljän silmän periaate on yksi mahdollisista toimenpiteistä, joita voidaan valvoa vaatimustenmukaisuustarkistuksen avulla. Tapahtumalokitetietojen ja ideaaliprosessimallin yhteensopivuuden tarkistaminen voi paljastaa mahdollisia prosessivariaatioita tai muita poikkeamia, jotka saattavat vaarantaa prosessien oikeellisuuden ja tehokkuuden. Vaatimustenmukaisuustarkistuksen suurena etuna on sen kyky havaita, paikantaa ja selittää prosessipoikkeamia sekä mitata näiden poikkeamien vaikutusta prosessien suorituskyykyyn (Jans ym. 2011; Van der Aalst 2011).

On kuitenkin tärkeää huomata, että monet todelliset prosessit eivät ole alun perin suunniteltuja tai optimoituja, vaan ne ovat kehittyneet ajan myötä ilman selkeää määrittelyä tai mallinusta. Tällaisissa tapauksissa vaatimustenmukaisuustarkistusta ei voida suoraan hyödyntää, mutta prosessilouhinnan avulla voidaan silti saavuttaa merkittäviä hyötyjä prosessien löytämisen (discovery) kautta. Tällaiset tilanteet ovat erityisen mielenkiintoisia, sillä ne eivät rajoitu ainoastaan jo tunnetun prosessimallin uudelleen löytämiseen, vaan tarjoavat mahdollisuuden paljastaa aiemmin piilotettua tietoa ja ymmärtää prosessien todellista luonnetta (Günther ja Van der Aalst 2007).

Kolmas prosessilouhinnan tyyppi on tehostaminen (enhancement), jonka tarkoituksena on parantaa olemassa olevaa prosessimallia hyödyntämällä tapahtumalokitetiedoista saatuja tietoja todellisista prosesseista. Tehostamisen päämääränä on päivittää ja laajentaa ideaalista prosessimallia niin, että se vastaa paremmin todellisuutta. Van der Aalst (2011) erottaa tehostamisen kahteen päätyyppiin: korjaukseen ja laajennukseen.

Korjauksen (repair) ideana on päivittää prosessimalli siten, että se heijastaa tarkemmin todellisia prosessinkulkuja. Esimerkiksi, jos prosessimallissa kaksi toimintaa on kuvattu tapahtuvaksi samanaikaisesti, mutta todellisuudessa ne aina tapahtuvat peräkkäin, prosessimallia voidaan korjata vastaamaan tätä todellista järjestystä.

Laajennus (extension) puolestaan tarkoittaa prosessimallin täydentämistä uusilla näkökulmil-
la, jotka saadaan tapahtumalokitiedoista. Tällaisia uusia näkökulmia voivat olla esimerkiksi
odotus- ja palveluajat, joiden avulla prosessin toimintojen suorituskykyä voidaan analysoida
tarkemmin. Laajennetun mallin avulla on mahdollista tunnistaa prosessien pullonkauloja ja
arvioida läpimenoaikoja, mikä tarjoaa arvokasta tietoa prosessien optimointiin (Sonawane ja
Patki 2015).



KUVA 4. Prosessilouhinnan päätyypit (Murillas ym. 2018)

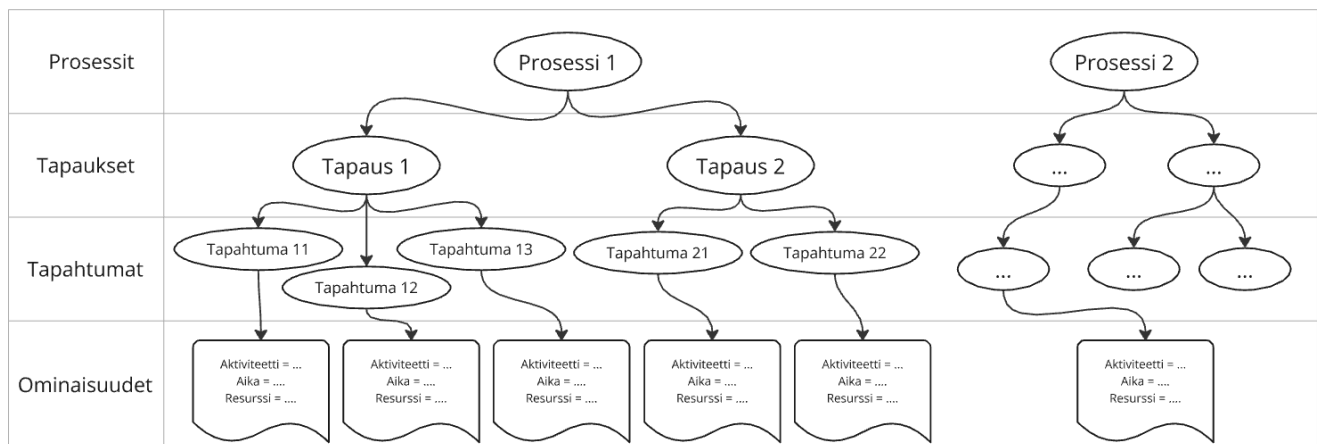
Edellä esiteltujen kolmen prosessilouhinnan päätyypin lisäksi Murillas ym. (2018) korostavat, että prosessilouhinta tarjoaa laajan valikoiman tekniikoita liiketoimintaprosessien parantami-
seen. Näihin tekniikoihin kuuluvat muun muassa vaatimustenmukaisuuden tarkistus, suori-
tuskyvyn analyysi, prosessien seuranta ja ennustaminen sekä toiminnallinen tuki. Esimerkiksi
Leontjeva, Conforti, Di Francescomarino sekä Dumas (2015) ovat tutkineet liiketoimintapro-
sessien ennakoivaa seurantaa, joka kuuluu prosessilouhinnan menetelmien luokkaan ja jon-

ka tavoitteena on ennustaa tapauksen ajoaika ja lopputulos mahdollisimman varhaisessa vaiheessa, perustuen epätäydellisiin tapahtumalokitietoihin. Vaikka nämä lisätekniikat tarjoavat erikoistuneita analyysimahdollisuuksia, ne voidaan kuitenkin yleisesti ottaen luokitella osaksi prosessilouhinnan päätyyppejä.

Prosessilouhinnan sovelluskohteet ovat moninaiset, ja sitä voidaan hyödyntää useista eri näkökulmista (Tiwari, Turner ja Majeed, 2017). Van der Aalst (2011) tunnistaa seuraavat keskeiset näkökulmat: ohjausvirta, organisaatio, tapaus ja aikaperspektiivi. Wu ym. (2019) huomauttavat, että ohjausvirtausnäkökulma on prosessilouhinnan tärkein näkökulma, mutta muut näkökulmat voivat tarjota syvempää analyysiä. Jans ym. (2011) käyttävät termiä "prosessinäkökulma" kuvaamaan ohjausvirtaa, koska se kattaa laajemmin tapahtumalokitietojen sisältämien toimintojen järjestyksen ja logiikan. Prosessinäkökulman tavoitteena on vastata selkeästi kysymykseen "Mitä polkuja seurataan?" ja visualisoida nämä polut esimerkiksi Petri-verkon, tapahtumaohjatun prosessiketjun (EPC) tai liiketoimintaprosessien mallinnusmerkinnän (BPMN) avulla (Jans ym. 2011).

2.1 Tapahtumalokit

Prosessilouhinnan perusta on tapahtumalokitiedot, jotka säilyvät keskeisenä käsitteenä prosessilouhinnassa. Sonawanan ja Patkin (2015), sekä Van der Aalstin (2011) mukaan jokainen tapahtuma tällaisessa lokissa viittaa tiettyyn toimintaan ja liittyy prosessitapaukseen. Tapahtumat tietyissä prosessitapauksissa tulee järjestää kronologisesti. Tapahtumalokit voivat sisältää myös lisätietoja tapahtumista, kuten henkilöstöresurssit ja aikaleimat. Näitä lisätietoja hyödynnetään edistyneemmässä prosessilouhinnassa, esimerkiksi sosiaalisten verkostojen analyysissä ja suorituskyvyn arvioinnissa. Kuvassa 5 esitetään esimerkkitapahtumalokin hierarkkinen rakenne. Esimerkissä on kuvattu kaksi prosessia, joista kummallakin on useita tapauksia. Tapaukset, eli prosessi-instanssit, edustavat käsiteltäviä kohteita, kuten asiakastilauksia tai työhakemuksia (Van der Aalst ym. 2007). Jokaisella tapauksella on tapahtumia, jotka on yksilöity tapahtuma avaimilla, kuten "11" tai "21", ja jokaisella tapahtumalla on kolme attribuuttia: toiminta, aikaleima ja resurssi. Aktiviteetit viittaavat yleensä tapauksen suorittamaan toimintaan, ja aikaleima ilmaisee ajankohdan, jolloin toiminta tapahtui.



KUVA 5. Esimerkki tapahtumalokien hierarkiasta.

Yleisemmässä kontekstissa tapahtumalokit koostuvat jäljitysjoukoista, joissa kukin jäljitys on tapahtumasarja, joka liittyy samaan tapaukseen. Attribuutteja voidaan liittää paitsi yksittäisiin tapahtumiin, myös koko tapahtumalokiin ja yksittäisiin jälkiin (Sonawane ja Patki 2015). Van der Aalst (2011) on määritellyt tapauksen, jäljityksen ja tapahtumalokin käsitteet seuraavasti: C on tapausten universumi, eli kaikkien mahdollisten tapaustunnisteiden joukko.

E on tapahtumien universumi, eli kaikkien mahdollisten tapahtumatunnisteiden joukko.

AN on attribuuttien nimien joukko.

A^* edustaa kaikkia äärellisiä sekvenssejä joukosta A .

Tapauksilla on ominaisuuksia. Jokaiselle tapaukselle $c \in C$ ja attribuuttinimelle $n \in AN$ pätee, että $\#n(c)$ on attribuutin n arvo tapauksessa c . Jos tapauksella c ei ole attribuuttia nimeltä n , niin $\#n(c)$ on tyhjä. Jokaisella tapauksella on erityinen pakollinen attribuutti jälki, merkittynä $\#trace(c)$, missä $\#trace(c) \in E^*$. (Oletamme jatkossa, että $\#trace(c)$ ei ole tyhjä, eli jokaisen tapauksen jälki sisältää vähintään yhden tapahtuman.) Merkintä c^\wedge viittaa tapauksen c jälkeen.

Jälki on äärellinen tapahtumasarja $\sigma \in E^*$, joka täyttää seuraavat ehdot:

$\sigma = (e_1, e_2, \dots, e_n)$, missä $e_i = \sigma(i)$ jokaiselle $1 \leq i \leq n$, ja n on tapahtumasekvenssin pituus.

Kukin tapahtuma esiintyy jäljessä vain kerran, eli jos $1 \leq i < j \leq n$, niin $\sigma(i) \neq \sigma(j)$.

Tapahtumaloki on tapausjoukko $L \subseteq C$ siten, että jokainen tapahtuma esiintyy enintään kerran koko lokissa. Toisin sanoen, mille tahansa $c_1, c_2 \in L$, jos $c_1 \neq c_2$, pätee:

$\partialset(c_1^\wedge) \cap \partialset(c_2^\wedge) = \emptyset$, missä ∂set muuntaa sekvenssin joukoksi. Esimerkiksi $\partialset(a, a, a, b, d) = \{a, b, d\}$.

Jos tapahtumaloki sisältää aikaleimoja, jälkien järjestyksen tulee noudattaa näitä aikaleimoja.

Tämä tarkoittaa, että jokaiselle $c \in L$ ja indekseille i ja j , joille pätee $1 \leq i < j \leq |c^\wedge|$:

- a. $|c^i|$ on tapauksen c jäljen pituus.
- b. $\#time(c^i) \leq \#time(c^j)$, missä $\#time(c^i)$ viittaa jäljen i tapahtuman aikaleimaan ja $\#time(c^j)$ jäljen j tapahtuman aikaleimaan.

Määritelmän ja kuvan 5 perusteella on huomioitava, että kukin tapahtuma voi kuulua vain yhteen tapaukseen, ja jokainen tapaus voi kuulua vain yhteen prosessiin. Tästä syystä sekä tapahtumat että tapaukset esitetään yksilöllisillä tunnisteilla. Tunniste $e \in E$ viittaa tapahtumaan, kun taas tunniste $c \in C$ viittaa tapaukseen. Tämän mekanismin avulla voimme tarkasti osoittaa tiettyyn tapahtumaan tai tapaukseen. Yksilölliset tunnisteet ovat välttämättömiä jokaiselle tapaukselle ja tapahtumalle, koska useilla tapahtumilla voi olla samat attribuutit, ja useat tapahtumat voivat liittyä yhteen tapaukseen.

Van der Aalst (2011) huomauttaa, että nämä tunnisteet toimivat ensisijaisesti viitteinä tiettyihin tapahtumiin tai tapauksiin, eikä niiden tarvitse olla peräisin alkuperäisestä tietolähteestä. Ne voidaan luoda dataa poimittaessa eri lähteistä. On tärkeää ymmärtää, että jokainen tapahtuma on tallennettu toiminnon suoritus, ja mitä tahansa toimintaa voidaan kutsua tapahtumaluokaksi (Sonawane ja Patki 2015).

Tapahtumalokien hankkiminen todellisessa ympäristössä ei ole yksinkertainen tehtävä, kuten Murillas ym. (2018) korostavat. Prosessilouhintaprojekteissa voidaan odottaa, että tapahtumalokit täytyy kerätä useista eri lähteistä, kuten ERP-järjestelmistä, erillisistä tiedostoista ja tietokannoista. Van der Aalst (2011) sekä Murillas ym. (2018) huomauttavat ongelmista, jotka liittyvät tapahtumalokietietojen yhdistämiseen ja poimimiseen. Useiden tapahtumalokien yhdistämisessä semantiikka ja syntaksi osoittautuvat tärkeiksi tekijöiksi. Esimerkiksi tapahtumalokit eivät usein ole oikeassa muodossa, tai sellaisia lokeja ei välttämättä ole lainkaan saatavilla. Van der Aalst ym. (2011) esittelevät manifestin, jossa datan laadun kypsyystasot on jaettu viiteen luokkaan yhden ja viiden tähden välillä. Yhden tähden taso edustaa alinta kypsyysastetta: tällaiset tapahtumalokit tallennetaan manuaalisesti ja voivat sisältää puuttuvia tai virheellisesti tallennettuja arvoja. Viiden tähden tapahtumalokit puolestaan edustavat korkeinta kypsyysastetta, mikä tarkoittaa, että järjestelmä tallentaa tapahtumalokit automaattisesti, ja ne ovat täydellisiä ja tarkkoja.

Prosessilouhinnan lähtökohtana on "tapahtumaloki," joka sisältää tietoa liiketoimintaprosesseista. Prosessilouhinnan ideana on poimia tapahtumalokeista tietoa liiketoimintaprosessien toiminnasta. Jokainen kerta, kun työntekijä suorittaa liiketoimintaprosessin, sitä pidetään prosessin esiintymänä eli tapauksena. Koska useat eri työntekijät voivat suorittaa saman liiketoimintaprosessin useita kertoja, ja jokainen heistä saattaa suorittaa prosessin eri tavalla, tapahtumalokiin kertyy useita prosessin esiintymiä (Verbeek ja Van der Aalst 2014).

Ensimmäinen tapahtumalokien määritelmä esitetään seuraavasti.

Määritelmä 1: Tapaus on liiketoimintaprosessin esiintymä.

Liiketoimintaprosessissa on yleensä useita toimintoja, jotka muodostavat prosessin tarkoin määritellyjä vaiheita, ja jotka suoritetaan tietyssä järjestyksessä. Nämä toiminnot on nimetty, ja ne kirjataan tapahtumalokiin suorituksen yhteydessä. Tapahtumalokiin kirjattava tieto sisältää usein enemmän kuin pelkän tapahtumajärjestyksen ja tehtävän otsikon; esimerkiksi suoritettujen toimintojen alkamis- ja päättymisajat, kustannukset sekä toiminnon suorittanut henkilö tai kone voivat kuulua tallennettaviin tietoihin. Tapahtumalokissa rekisteröityjä toimintoja kutsutaan tapahtumiksi, ja jokainen tapaus muodostuu tapahtumasarjasta, jossa jokainen tapahtuma vastaa täsmällisesti yhtä tapausta.

Määritelmä 2: Tapahtuma on liiketoimintaprosessissa suoritettu toiminto.

Lisäksi on tärkeää määritellä "jäljitys" (trace), joka kuvaa tapahtumien sekvenssiä, joka voi esiintyä yhdessä tai useammassa tapauksessa tapahtumalokissa. Koska liiketoimintaprosessi suoritetaan useita kertoja, on mahdollista, että sama tapahtumajärjestys toistuu useissa eri tapauksissa. Näin ollen jäljitys on joukko erillisiä aktiviteettisekvenssejä, jotka esiintyvät useammassa kuin yhdessä tapauksessa.

Määritelmä 3: Jäljitys on tapahtumasarja, joka esiintyy yhdessä tai useammassa tapauksessa tapahtumalokissa.

Tapahtumalokien tiedot poimitaan yleensä tietokannoista, ja ne edustavat vain tilannekuvaa todellisista prosesseista. Yleisesti ottaen prosessitietojen keräämiseen liittyy kolme oletusta:

Jokainen tapahtuma viittaa tiettyyn tehtävään.

Jokainen tapahtuma viittaa tiettyyn tapaukseen.

Tapahtumat ovat täysin järjestettyjä.

Taulukko 1 esittelee esimerkinomaisen tapahtumalokin.

TAULUKKO 1. Esimerkki tapauslokin datasta

Tapaus ID	Tapahtuma	Resurssi	Aloitusaika	Lopetusaika
1000184	Arvioi hakemus	Team X	24/01/2015 14.36	26/01/2015 14.52
1000184	Odotus	NULL	26/01/2015 14.52	NULL
1000184	Kerää aineistoa	Team X	28/01/2015 13.47	28/01/2015 13.48
1000185	Arvioi hakemus	Team Y	25/01/2015 14.36	28/01/2015 14.52

2.2 Prosessimallit

Prosessilouhinta on osittain prosessimallien johdattamista syötteenä annetuista tapahtumalokeista. Prosessimalli kuvaa taustalla olevaa liiketoimintaprosessia, ja näiden mallien esittämiseen on olemassa monia erilaisia merkintätapoja, jotka voivat olla joko visuaalisia tai tekstimuotoisia. Tämän työn kannalta on tärkeää valita visuaalinen merkintätyyppi. Valinnan perustana on se, että työn tuloksia arvioivat ihmiset, joilla on erilaisia taustoja, ja ilman syvälistä tietoa tekstipohjaisten mallien ymmärtäminen voi olla haastavaa. Visuaaliset mallit ovat yleensä intuitiivisempia ja siksi helpompia ymmärtää.

Yksi yleisimmin käytetyistä visuaalisten mallien merkintätavoista on niin sanottu 'Petri-verkko' (Petri Net), joka on erityisen sopiva liiketoimintaprosessien kuvaamiseen ja jota käytetään myös tässä työssä (Aalst, Weijters & Maruster 2002).

Petri-verkko määritellään muodollisesti kolmioksi (P, T, F) , jossa:

P on äärellinen joukko paikkoja (places).

T on äärellinen joukko siirtymiä (transitions) siten, että $P \cap T = \emptyset$ (eli paikkojen ja siirtymien joukot ovat toisistaan erillisiä).

$F \subseteq (P \times T) \cup (T \times P)$ on äärellinen joukko suunnattuja kaaria, joita kutsutaan virtaussuhteiksi (flow relations).

Petri-verkko N voidaan merkitä merkinnällä s , jolloin tuloksena on merkitty Petri-verkko (N, s) . Merkintä s tarkoittaa joukkoa paikkoja, joihin liittyy tokeniksi kutsuttuja elementtejä. Merkityissä Petri-verkoissa siirtymiä voidaan "ampua" (fire), mikä tarkoittaa, että siirtymän aktivointi muuttaa Petri-verkon merkintää. Jotta siirtymän ampuminen olisi mahdollista, on ensin määriteltävä tulo- ja lähtöpaikat.

2.2.1 Prosessin visualisointi

Jotta voidaan havainnollistaa prosessilouhinnan periaatteita, tarkastellaan taulukossa 1 esitettyä työnkulukokia. Tämä loki sisältää tietoja viidestä tapauksesta, eli työnkulun ilmentymästä. Lokista ilmenee, että neljässä ensimmäisessä tapauksessa (1, 2, 3 ja 4) on suoritettu tehtävät A, B, C ja D. Viidennessä tapauksessa sen sijaan on suoritettu vain kolme tehtävää: A, E ja D.

Huomionarvoista on, että jokainen tapaus alkaa tehtävän A suorituksella ja päättyy tehtävään D. Lisäksi havaitaan, että jos tehtävä B suoritetaan, niin myös tehtävä C suoritetaan, mutta niiden suoritusjärjestys voi vaihdella; joissakin tapauksissa C suoritetaan ennen B:ä. Viidennessä tapauksessa tehtävät B ja C ohitetaan, ja niiden sijaan suoritetaan tehtävä E ennen siirtymistä päätösvaiheeseen D.

Taulukossa 1 esitettyjen tietojen perusteella ja olettaen lokin täydellisyyden eli että tapaukset ovat edustavia ja kattavat riittävän suuren osan mahdollisista käyttäytymismalleista voimme päätellä prosessimallin, joka kuvaa havaittua toimintaa. Tämä prosessimalli on esitetty kuvassa 6 Petri-verkon avulla (van der Aalst, 2011).

TAULUKKO 2. Työnkulun loki

Tapausnumero	Tehtävän tunnus
tapaus 1	Tehtävä A
tapaus 2	Tehtävä A
tapaus 3	Tehtävä A
tapaus 3	Tehtävä B
tapaus 1	Tehtävä B
tapaus 1	Tehtävä C
tapaus 2	Tehtävä C
tapaus 4	Tehtävä A
tapaus 2	Tehtävä B
tapaus 2	Tehtävä D
tapaus 5	Tehtävä A
tapaus 4	Tehtävä C
tapaus 1	Tehtävä D
tapaus 3	Tehtävä C
tapaus 3	Tehtävä D
tapaus 4	Tehtävä B
tapaus 5	Tehtävä E
tapaus 5	Tehtävä D
tapaus 4	Tehtävä D

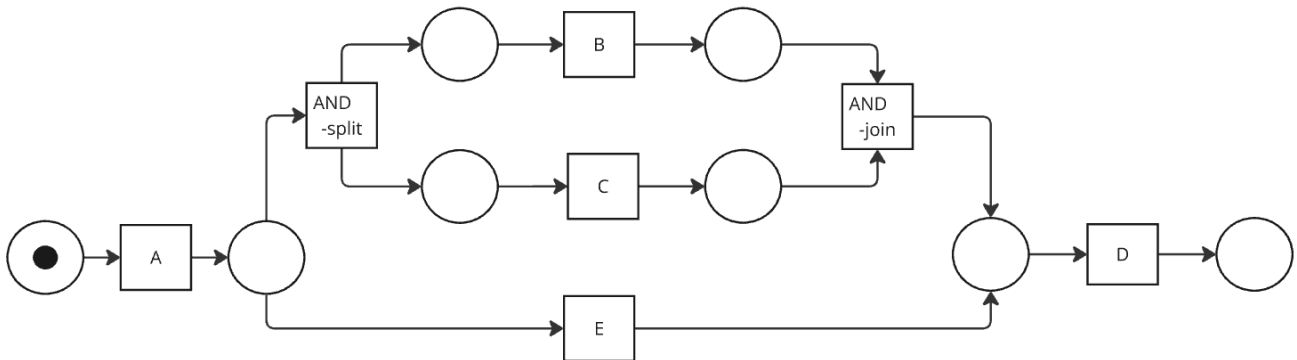
Petri-verkossa prosessi visualisoidaan paikkojen ja siirtymien verkostona, jossa voidaan mallintaa rinnakkaisuutta, valintaa ja sekvensointia. Kuvan 6 mallista nähdään, että prosessi alkaa tehtävällä A, jonka jälkeen tapahtuu haarautuminen. Haarautumisen jälkeen voidaan joko suorittaa tehtävät B ja C joko järjestyksessä B sitten C tai C sitten B tai vaihtoehtoisesti suoraaan tehtävä E. Lopuksi kaikki polut johtavat tehtävään D, joka päättää prosessin.

Tällainen mallinnus auttaa ymmärtämään prosessin rakenteen ja mahdolliset variaatiot, mikä on keskeistä prosessilouhinnan tavoitteiden kannalta. Se mahdollistaa myös prosessin optimoinnin ja pullonkaulojen tunnistamisen, kun tiedämme, miten eri tapaukset etenevät prosessissa.

Petri-verkko käynnistyy tehtävällä A ja päättyy tehtävään D, joita molempia edustavat siirtymät verkossa. A

suorittamisen jälkeen on mahdollista valita joko tehtävien B ja C suorittaminen rinnakkain tai pelkästään tehtävän E suorittaminen. Rinnakkaisen suorittamisen mahdollistamiseksi B ja C

on lisätty kaksi näkymätöntä tehtävää, nimeltään AND-split ja AND-join. Nämä lisäykset palvelevat ainoastaan reititystarkoituksia eivätkä näy työnkulun lokissa.



Prosessimalli työnkulun mukaisesti Van der Aalst

KUVA 6 Prosessimalli (Van der aalst 2011)

On tärkeää huomata, että voidaan olettaa kahden tehtävän olevan rinnakkaisia, jos ne voivat esiintyä missä tahansa järjestyksessä. Kuitenkin, jos erotellaan tehtävien aloitus- ja lopetustapahtumat, voimme eksplisiittisesti havaita rinnakkaisuuden. Aloitus- ja lopetustapahtumia voidaan myös hyödyntää osoittamaan, että tehtävien suorittaminen vie aikaa.

Esityksen yksinkertaistamiseksi oletamme kuitenkin tässä yhteydessä, että tehtävät ovat atomisia, menettämättä kuitenkaan yleisyyttä. Oletamme yksityiskohtaisen tapahtumamallin tehtäville, johon sisältyy tapahtumia kuten "aloita tehtävä", "peruuta tehtävä", "jatka tehtävää" ja "suorita tehtävä loppuun". On kuitenkin oleellista ymmärtää, että tällainen lähestymistapa on tehokas vain, jos tapahtumat kirjataan tarkasti niiden tapahtumahetkellä.

2.3 Prosessilouhinnan hyödyntäminen ERP:in yhteydessä

Nykyaikaiset työnkulun hallintajärjestelmät perustuvat eksplisiittisiin prosessimalleihin, mikä tarkoittaa, että täysin määritelty työnkulkumalli on välttämätön tietyn työnkulkuprosessin toteuttamiseksi. Työnkulkumallin luominen on kuitenkin monimutkainen ja aikaa vievä prosessi, ja usein todellisten työnkulkuprosessien ja johdon havaitsemien prosessien välillä on eroja. Tämän haasteen ratkaisemiseksi on kehitetty tekniikoita työnkulkumallien automaattiseen löytämiseen.

Näiden tekniikoiden perustana on niin sanottu "työnkulkuloki," joka sisältää tietoa siitä, miten työnkulkuprosessit todellisuudessa suoritetaan. (Aalst ym.) esittävät uuden algoritmin, jonka avulla prosessimalli voidaan poimia työnkulkulokista ja esittää se Petri-verkon avulla. He osoittavat kuitenkin myös, että kaikkia mielivaltaisia työnkulkuprosesseja ei ole mahdollista löytää. Tässä työssä keskitymme tutkimaan työnkulkuprosessien luokkaa, jotka voidaan luotettavasti löytää.

Työnkulun hallintajärjestelmät, kuten Staffware, IBM MQSeries, COSA jne., tarjoavat yleisiä mallinnus- ja käyttöönottovalmiuksia strukturoiduille liiketoimintaprosesseille. Tekemällä graafisia prosessimääritelmiä eli malleja, jotka kuvaavat tyypillisen tapauksen (työnkulun ilmentymän) elinkaaren erikseen, nämä järjestelmät voidaan konfiguroida tukemaan liiketoimintaprosesseja. Puhtaiden työnkulun hallintajärjestelmien lisäksi monet muut ohjelmistojärjestelmät ovat omaksuneet työnkulkutekniikkaa.

Esimerkiksi ERP (Enterprise Resource Planning) -järjestelmä, kuten Business Central, SAP tai CRM (Customer Relationship Management) -ohjelmistot jne. tarjoavat nykyään samat mahdollisuudet. Lupauksestaan huolimatta työnkulkuteknologiaa käytettäessä kohdataan monia ongelmia. Yksi ongelmista on, että nämä järjestelmät vaativat työnkulun suunnittelun, eli suunnittelijan on rakennettava yksityiskohtainen malli, joka kuvaa tarkasti työn reitityksen. Työnkulun mallintaminen ei ole läheskään triviaalia: se vaatii syvällistä tietoa järjestelmän työnkulusta ja pitkiä keskusteluja mukana olevien työntekijöiden ja johdon kanssa.

3 KÄYTÄNNÖN TOTEUTUS

Tämän työn ensisijaisena tavoitteena oli syvällisesti tutkia prosessilouhinnan tarjoamia mahdollisuuksia sekä tunnistaa ja analysoida erilaisia prosessikokonaisuuksia. Halusimme erityisesti tarkastella näiden prosessien eri ilmentymiä hyödyntäen tehokasta visualisointia. Visualisoinnin avulla pyrimme tekemään monimutkaisista prosessivirroista helpommin ymmärrettäviä ja analysoitavia, mikä on kriittistä liiketoimintaprosessien optimoinnissa.

Työn alkuvaiheessa suoritin perusteellisen suunnitteluvaiheen, jossa selvitin erilaisia tapoja saada tarvittava data ulos Business Central -järjestelmästä sellaisessa muodossa, että sitä olisi helppo louhia ja analysoida. Tämä sisälsi datan rakenteen ja muodon tarkastelua sekä sitä, miten data voitaisiin parhaiten muuntaa prosessilouhintaan sopivaksi. Tavoitteena oli varmistaa, että käytössämme oleva data olisi mahdollisimman laadukasta ja käyttökelpoista jatkoanalyysiä varten.

Microsoft Dynamics 365 Business Centralin telemetrian hyödyntäminen ja datan tuominen Azure Application Insightsiin on nykyaikainen ratkaisu, joka mahdollistaa yrityksille syvällisen näkemyksen liike-toimintaprosessiensa tehokkuudesta ja suorituskyvystä. Tässä käydään läpi, kuinka tämä integraatio voidaan toteuttaa ja hyödyntää prosessilouhinnassa, keskittyen erityisesti tekniseen toteutukseen ja siihen, miten data voidaan sisällyttää telemetriaan (Demiliani ja Tacconi 2024).

1. Vaihe: Telemetrian konfigurointi Business Centralissa

Ensimmäinen askel on konfiguroida Microsoft Dynamics 365 Business Central lähettämään telemetriotietoja. Business Central tarjoaa sisäänrakennetun tuen telemetrian lähettämiseen Azure Application Insightsiin, mikä tekee integraation aloittamisesta suoraviivaista. Käyttäjän on luotava Azure Application Insights -resurssi Azure-portaalissa ja kopioitava sovelluksen Insights-avain. Tämä avain syötetään sitten Business Centralin ympäristöasetuksiin, mikä oikeuttaa Business Centralin lähettämään telemetriotietoja suoraan Azureen.

2. Vaihe: Attribuuttien määrittely ja lähetys

Tärkeä osa integraatiota on määrittellä, mitä tietoja halutaan seurata. Business Centralin telemetriaan sisällytetään vakiotiedot, kuten suorituskymittarit ja käyttötilastot, mutta yritykset

voivat hyödyntää myös kustomoitua telemetriaa. Tämä tarkoittaa attribuuttien, kuten nimikkeen numeron, määrittämistä ja niiden sisällyttämistä telemetriavirtoihin. Kehittäjät voivat käyttää AL-kieltä (Business Centralin kehityskieli) kustomoitujen telemetriatapahtumien lähettämiseen, määrittäen tarvittavat attribuutit ja arvot, jotka kuvaavat liiketoimintaprosesseja tarkemmin.

3. Vaihe: Datan siirto Azure Application Insightsiin

Kun Business Central on konfiguroitu lähettämään telemetriatietoja ja attribuutit on määritelty, tiedot alkavat virrata automaattisesti Azure Application Insightsiin. Azure Application Insights kerää ja analysoi näitä tietoja reaaliajassa, tarjoten visuaalisia raportteja ja hälytyksiä suorituskyvystä, käytöstä ja mahdollisista ongelmista. Käyttäjät voivat mukauttaa näkymiä ja luoda omia raporttejaan vastaamaan yrityksen tarpeita.

4. Vaihe: Prosessilouhinta ja analyysi

Kerättyä telemetriatietoa voidaan hyödyntää prosessilouhinnassa, joka on menetelmä liiketoimintaprosessien analysointiin ja kehittämiseen datan avulla. Prosessilouhinta Azure Application Insightsin datalla mahdollistaa syvälliset analyysit prosessien tehokkuudesta, pullonkaloista ja parannusmahdollisuuksista. Attribuutit, kuten nimikkeen numero, mahdollistavat tarkemman analyysin tiettyjen prosessien tai toimintojen tehokkuudesta.

Integraation avulla Business Centralin telemetriatiedot siirretään sujuvasti Azure Application Insightsiin, tarjoten yrityksille arvokasta tietoa liiketoimintaprosessiensa suorituskyvystä ja mahdollisista kehityskohteista. Tekninen toteutus keskittyy telemetrian konfigurointiin, attribuuttien määrittelyyn, datan siirtoon ja lopulta prosessilouhintaan, mikä mahdollistaa dataveitoisen päätöksenteon ja prosessien jatkuvan parantamisen.

Demiliani ja Tacconi (2024) käsittelee asiaa, jossa telemetriaa hyödynnetään sovellusten suorituskyvyn mittaamiseen, käytön seuraamiseen, ongelmien diagnosointiin sekä käyttäjäkokemuksen parantamiseen. Tietoja voidaan lähettää Azure Application Insightsiin käyttämällä tälle luotua avainkoodia. Tämä mahdollistaa telemetriatietojen hyödyntämisen KQL:n avulla ja välittämisen myöskin prosessilouhintaa varten.

3.1 Datamallin suunnittelu

Datamalli toimii prosessilouhinnan perustana, kuten kappaleessa 2.1 kuvattiin. Tämä on kriittinen tekijä, kun lähdetään suunnittelemaan datamallin toteutusta niin sanotusti tyhjältä pohjalta. Datamallin suunnittelussa huomioitiin myös Business Centralin olemassa olevat mahdollisuudet datan tuottamiseen. Tarkastelussa on kaksi vaihtoehtoa, hyödyntää telemetriaa tai tehdä ratkaisu täysin räätälöitynä.

3.1.1 Datan tuottaminen telemetrian avulla

Hyödyntämällä telemetriaa datan tuottamiseen on tarpeen lisätä jokaiseen funktioon triggeri, joka leimaa tapahtumaan liittyvän tiedon ja lähettää sen Azure Application Insightsiin. Azure Application Insightsista voimme tehdä kyselyjä Kusto Query Languagella (KQL), jonka avulla voimme muodostaa tarvittaessa .csv-tiedoston. Tätä tiedostoa voidaan hyödyntää prosessilouhinnassa sekä visualisoida prosessikokonaisuuksia.

Haittapuolena tässä lähestymistavassa on kuitenkin se, että datamallin kyvykkyydet ovat rajoitetut telemetrian tarjoamiin tietoihin. Esimerkkinä telemetrian käyttämä metodi on seuraava:

```
Session.LogMessage(EventId: String, Message: String, Verbosity: Verbosity, DataClassification: DataClassification, TelemetryScope: TelemetryScope, Dimension1: String, Value1: String [, Dimension2: String, Value2: String])
```

Parametrit:

EventId: String

Yksilöllinen tunniste tapahtumalle tai viestille. Tämä erottaa tapahtumat toisistaan lokissa ja telemetriassa.

Message: String

Varsinainen viesti, joka halutaan kirjata. Se voi sisältää tarkentavia tietoja tapahtumasta.

Verbosity: Verbosity

Määrittää viestin tärkeysasteen tai yksityiskohtaisuuden tason, näitä arvoja voivat olla: Normal, Verbose, VeryVerbose.

DataClassification: DataClassification

Ilmoittaa viestin sisältämän datan luokittelun, tietosuojaan sekä tietoturvan näkökulmasta. Arvoja voivat olla: CustomerContent, EndUserIdentifiableInformation, SystemMetadata. Nämä auttavat hallitsemaan, kenelle telemetria tiedot ovat näkyvissä.

Dimension1: String, Value1: String

Mukautettu ulottuvuus ja arvo. Tätä voidaan hyödyntää lisätietojen lisäämisessä viestiin, esimerkiksi käyttämällä asiakasnumeroa, tilausnumeroa tai muuta vastaavaa tarkentavaa tunnistetta.

Dimension2: String, Value2: String

Toinen mukautettu ulottuvuus ja arvo. Tätä voidaan hyödyntää lisätietojen lisäämisessä viestiin, esimerkiksi käyttämällä asiakasnumeroa, tilausnumeroa tai muuta vastaavaa tarkentavaa tunnistetta.

Esimerkki käytännöstä:

```
Session.LogMessage(EventId: 'ORD001',Message: 'Uusi tilaus luotu.',Verbosity::Normal,DataClassification::SystemMetadata,TelemetryScope::ExtensionPublisher,'CustomerId', 'CUST001','OrderId', 'ORD12345');
```

Tämä metodi mahdollistaa tiettyjen tapahtumatietojen tallentamisen, mutta se on melko rajoitettu. Voimme tallentaa vain ennalta määritellyt kentät, mikä saattaa olla riittämätöntä monimutkaisempien prosessitietojen keräämiseen.

Metodin mukaisesti tiedot lähetetään suoraan Microsoftin Application Insightsiin. Data on hyödynnettävissä semmoisenaan KQL-kyselyllä, esimerkiksi ADX:ää käyttämällä. KQL:n peruskyselyn avulla voimme hakea tapahtumat ja muodostaa niistä esimerkiksi .csv-tiedoston. Alla on esimerkki KQL-kyselystä, joka hakee yhden käyttäjän istunnon sivukäynnit (Microsoft, 2024):

```
let _user_Id = 'b99ed22a-4681-41e8-b5fc-91c004e1979a';
let pv =
    pageViews
    | where user_Id == _user_Id
    | extend message = strcat('Page opened: ', name)
    | project timestamp, session_Id, user_Id, message, clientType='WebClient';
let tra =
    traces
```



```

| where user_Id == _user_Id
| project timestamp, session_Id, user_Id, message, clientType =
tostring(customDimensions.clientType);
union pv, tra
| order by timestamp asc

```

Tämä kysely hakee sekä sivukatselut "PageViews" että tapahtumat "traces" tietyn käyttäjän osalta ja yhdistää ne yhdeksi aikajärjestykseen järjestetyksi tulokseksi. Näin saamme kattavan kuvan käyttäjän toiminnasta järjestelmässä, mistä voisi olla hyötyä prosessilouhinnan kannalta.

Vaikka KQL tarjoaa tehokkaat työkalut datan kyselyyn ja käsittelyyn, olemme edelleen rajoitettuja telemetrian keräämän datan määrään ja laatuun. Telemetria mahdollistaa perusasioden tunnistamisen, mutta se ei välttämättä riitä syvällisempään analyysiin tai räätälöityjen datatarpeiden täyttämiseen. Esimerkiksi voimme kohdata haasteita, jos haluamme seurata monimutkaisia prosessivirtoja tai kerätä erityisiä liiketoimintatietoja, joita telemetria ei oletusarvoisesti tallenna.

Lisäksi telemetrian kautta kerätyn datan muokkaaminen ja laajentaminen voi olla haastavaa. Jos tarvitsemme lisää dimensioita tai erityisiä tietokenttiä, telemetrian tarjoamat mahdollisuudet voivat olla liian rajoitetut. Tämä voi johtaa siihen, että emme saa kerättyä kaikkea tarvittavaa dataa prosessilouhinnan tehokasta toteuttamista varten.

Telemetrian hyödyntäminen tarjoaa nopean ja suhteellisen helpon tavan kerätä perusdataa prosessilouhintaa varten. KQL:n avulla voimme tehdä monipuolisia kyselyjä ja saada hyödyllistä tietoa järjestelmän käytöstä. Kuitenkin telemetrian rajoitukset datamallin joustavuudessa ja kerättävän datan syvyydessä voivat muodostua esteiksi syvällisemmälle analyysille.

Näiden rajoitusten vuoksi on tarpeen harkita vaihtoehtoisia ratkaisuja, kuten oman taulun luomista. Tämä mahdollistaisi joustavamman ja räätälöidymmän datan keräämisen suoraan eri funktioista prosessivirtojen mukaisesti. Oman taulun avulla voimme määritellä tarkasti, mitä tietoja kerätään ja miten niitä käsitellään, mikä voi olla ratkaisevaa prosessilouhinnan onnistumisen kannalta.

On tärkeää punnita telemetrian ja oman datan keräämisen etuja ja haittoja projektin tavoitteiden ja resurssien näkökulmasta. Telemetria tarjoaa nopean aloituksen ja valmiit työkalut, mutta sen rajoitukset voivat estää syvällisemmän analyysin. Oman taulun luominen vaatii enemmän työtä ja resursseja, mutta se tarjoaa suuremman joustavuuden ja mahdollisuuden kerätä juuri haluttua dataa.

3.1.2 Datan tuottaminen räätälöidyn ratkaisukokonaisuuden avulla

Prosessilouhinnan tarpeisiin räätälöity ratkaisukokonaisuuden suunnittelu on huomattavasti joustavampi ja muokattavissa oleva ratkaisu. Suunnittelun lähtökohtana on enemmänkin tarpeiden kartoitus ja tämän jälkeen taulurakenteen luominen tarpeisiin nähden. Voimme tehdä helposti oman taulurakenteen ja lisätä siihen prosessivaiheista dataa prosessin mukaisesti. Alla oleva esimerkki taulurakenteen luomisesta AL-koodilla (Microsoft 2024).

```
table 50104 "Process Event Log"
{
    Caption = 'Process Event Log';
    DataPerCompany = true;

    fields
    {
        field(1; "Event ID"; Integer)
        {
            Caption = 'Event ID';
            AutoIncrement = true;
        }
        field(2; "Case ID"; Code[20])
        {
            Caption = 'Case ID';
        }
        field(3; "Activity Name"; Text[100])
        {
            Caption = 'Activity Name';
        }
    }
}
```

```

field(4; "Timestamp"; DateTime)
{
    Caption = 'Timestamp';
}
field(5; "User ID"; Code[50])
{
    Caption = 'User ID';
}
field(6; "Additional Info"; Text[250])
{
    Caption = 'Additional Information';
}
}

keys
{
    key(PK; "Event ID")
    {
        Clustered = true;
    }
    key("Case ID Key"; "Case ID")
    {
    }
}
}

```

Taulun rakenne ja kenttien selitykset

1. Event ID: Tämä on tapahtuman yksilöllinen tunniste. `AutoIncrement`-ominaisuus varmistaa, että jokaiselle tapahtumalle luodaan automaattisesti uniikki ID.
2. Case ID: Prosessiin tai tapaukseen liittyvä tunniste. Tämä mahdollistaa tapahtumien ryhmittelyn prosessikohtaisesti, mikä on keskeistä prosessilouhinnassa.

3. Activity Name: Tapahtuman tai aktiviteetin nimi. Tämä kenttä tallentaa, mitä toimintoa tapahtuma edustaa prosessissa.

4. Timestamp: Aikaleima, joka kertoo tarkasti, milloin tapahtuma on tapahtunut. Tämä on olennainen tieto tapahtumien aikajärjestyksen analysoinnissa.

5. User ID: Käyttäjän tunniste, joka suoritti tapahtuman. Tämä mahdollistaa resurssien käytön analysoinnin ja prosessin läpiviennin seurannan käyttäjätasolla.

6. Additional Info: Mahdolliset lisätiedot tapahtumasta. Tämä kenttä tarjoaa joustavuutta tallentaa muita tarpeellisia tietoja, jotka voivat olla hyödyllisiä analyysissä.

Taulun avaimet

- PK (Primary Key): Pääavain on "Event ID", joka takaa jokaisen tapahtuman yksilöllisyyden.
- Case ID Key: Toinen avain "Case ID" mahdollistaa nopean haun ja ryhmittelyn prosessikohtaisesti.

Jotta voimme kerätä dataa tähän taulun, käytämme niin kutsuttuja tapahtumatilaaajia tai kuuntelijoita (EventSubscribers). Tapahtumatilaaajat "kuuntelevat" järjestelmän tapahtumia ja suorittavat määritellyn koodin, kun tietty tapahtuma tapahtuu järjestelmässä. Näin voimme automaattisesti tallentaa prosessien tapahtumat tauluun.

Esimerkki koodiyksiköstä (Microsoft 2024), joka tallentaa tapahtumat

codeunit 50104 "Process Event Subscriber"

```
{
    [EventSubscriber(ObjectType::Table, Database::"Sales Header", 'OnAfterInsertEvent', "",
true, true)]
    local procedure OnAfterSalesHeaderInsert(Rec: Record "Sales Header")
    var
        ProcessEventLog: Record "Process Event Log";
    begin
        ProcessEventLog.Init();
        ProcessEventLog."Case ID" := Format(Rec."No.");
```

```

    ProcessEventLog."Activity Name" := 'Sales Order Created';
    ProcessEventLog."Timestamp" := CurrentDateTime();
    ProcessEventLog."User ID" := UserId();
    ProcessEventLog.Insert();
end;

```

```

[EventSubscriber(ObjectType::Table, Database::"Sales Header", 'OnAfterModifyEvent', "",
true, true)]

```

```

local procedure OnAfterSalesHeaderModify(Rec: Record "Sales Header")
var
    ProcessEventLog: Record "Process Event Log";
begin
    ProcessEventLog.Init();
    ProcessEventLog."Case ID" := Format(Rec."No.");
    ProcessEventLog."Activity Name" := 'Sales Order Modified';
    ProcessEventLog."Timestamp" := CurrentDateTime();
    ProcessEventLog."User ID" := UserId();
    ProcessEventLog.Insert();
end;

```

Selitys koodista:

codeunit 50104 "Process Event Subscriber": Koodiyksikkö, joka sisältää tapahtumatilaaajat.

[EventSubscriber(...)] : Määrittää tapahtuman, jota tilaaja kuuntelee. Esimerkissä kuuntelemme myyntitilauksen (`Sales Header`) lisäämistä ja muokkausta.

OnAfterSalesHeaderInsert ja OnAfterSalesHeaderModify: Menetelmät, jotka suoritetaan tapahtumien jälkeen.

ProcessEventLog: Viittaus luomaamme tauluun, johon tallennamme tapahtumatiedot.

ProcessEventLog.Init(): Alustaa uuden tietueen.

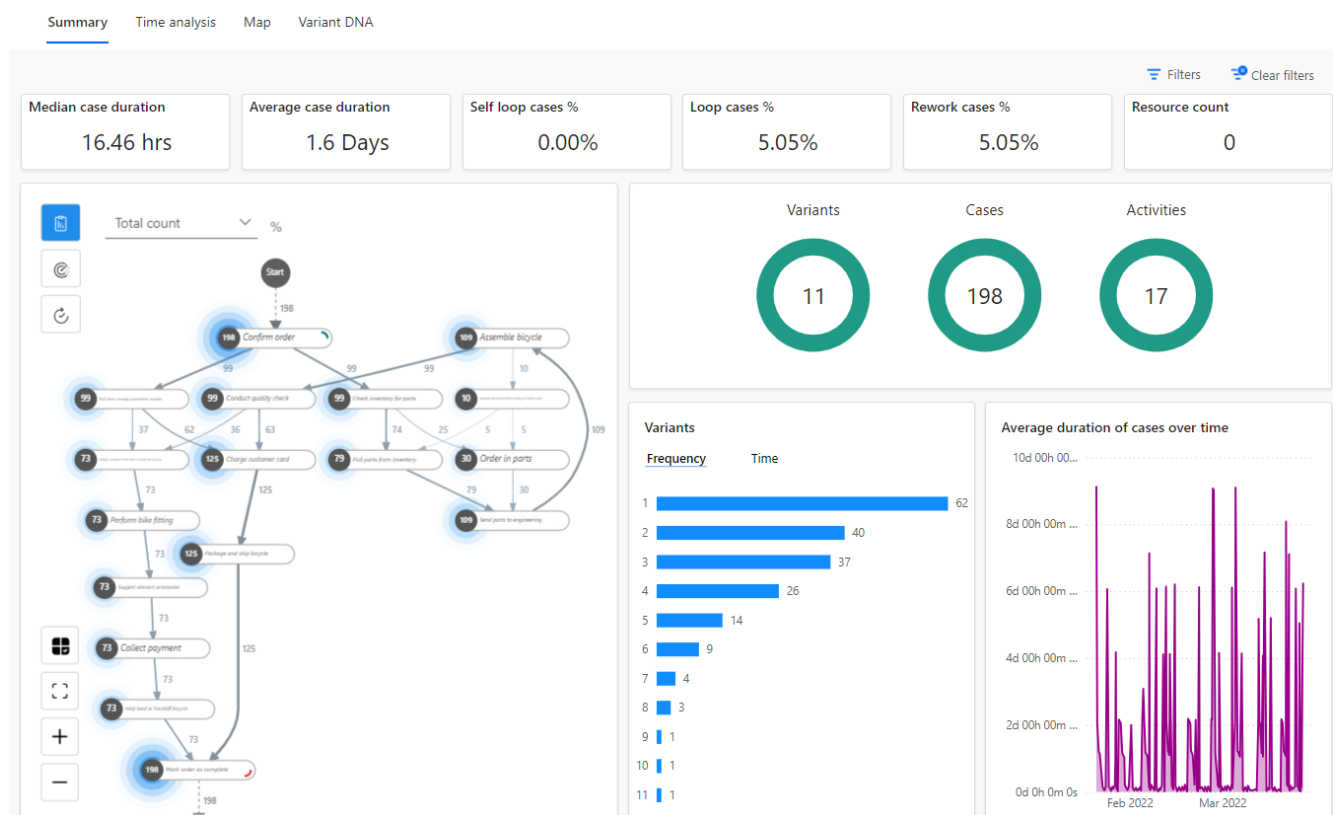
ProcessEventLog.Insert(): Tallentaa tietueen tauluun.

Luotuamme taulun ja kerättyämme dataa voimme viedä tiedot analysointia varten. Tämä voidaan tehdä esimerkiksi seuraavilla tavoilla:

1. Excel-vienti: Luodaan sivu, joka näyttää "Process Event Log" -taulun tiedot, ja käytämme Business Centralin sisäänrakennettua Excel-vientitoimintoa muodostaakseen .csv tiedosto aineistosta.
2. Dataportit tai raportit: Voimme luoda dataportin tai raportin, joka vie tiedot haluttuun muotoon, kuten CSV-tiedostoksi.
3. Web-palvelut: Julkaisemme taulun web-palveluna, jolloin ulkoinen prosessilouhintatyökalu voi hakea tiedot suoraan tietokannasta kyselyn avulla.

3.1.3 Datan analysointi

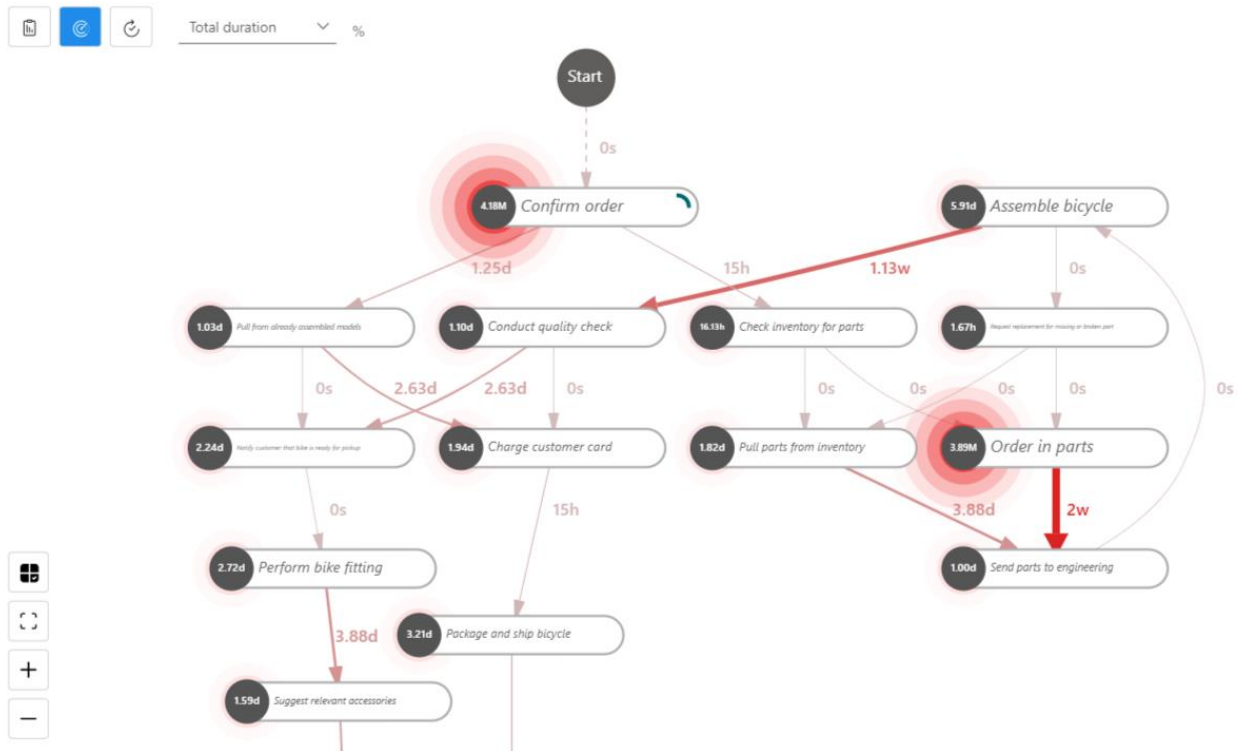
Datan analysoimiseen ja prosessilouhintaan on nykyään saatavilla lukuisia eri sovelluksia ja valmiita ratkaisuja, jotka auttavat organisaatioita ymmärtämään ja optimoimaan liiketoimintaprosessejaan. Näiden työkalujen avulla voidaan kerätä, käsitellä ja visualisoida suuria määriä prosessidataa, mikä mahdollistaa syvällisen analyysin ja tehokkaiden kehitystoimenpiteiden suunnittelun. Tässä työssä hyödynsin kahta keskeistä työkalua: Microsoftin Process Mining -työkalua sekä pm4py:tä, joka on Python-ohjelmointikieleen perustuva avoimen lähdekoodin projekti.



Kuva 7 Microsoft Process Mining -työkalu

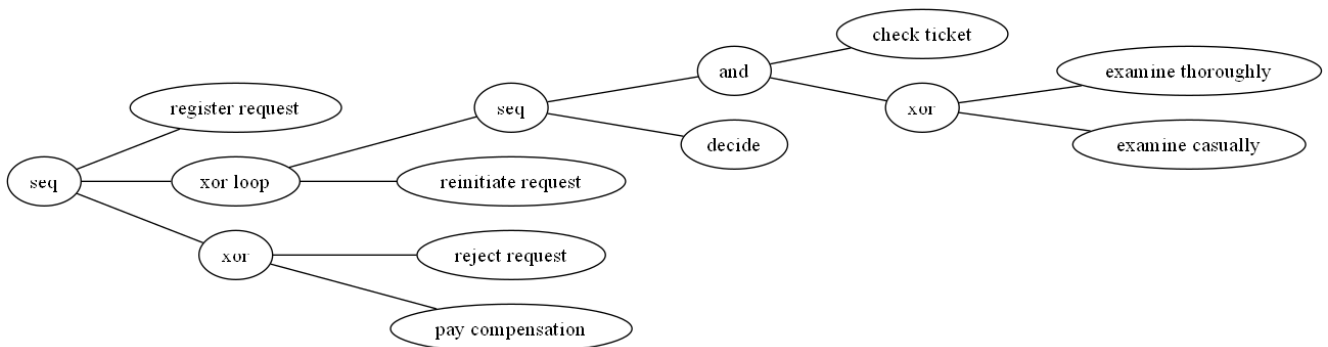
Microsoftin Process Mining -työkalu on integroitu osa Microsoftin Power BI -ympäristöä, mikä tekee siitä helposti lähestyttävän ja laajasti käytettävissä olevan ratkaisun prosessilouhintaan. Työkalu tarjoaa käyttäjäystävällisen käyttöliittymän, jonka avulla voidaan helposti ladata ja käsitellä dataa eri lähteistä, kuten Business Centralista, ja visualisoida prosessivirtoja interaktiivisilla grafiikoilla. Kuvassa 8 esitetään prosessivirran tehokkuus ja mahdolliset tehokkuuteen liittyvät huomiot, kuten "Confirm order" sekä "Order in parts" ovat korostettuina. Ko-

rostus tarkoittaa tässä tapauksessa sitä, että ne ovat 2 prosessivaihetta, jotka kestävät pi-
simpään.



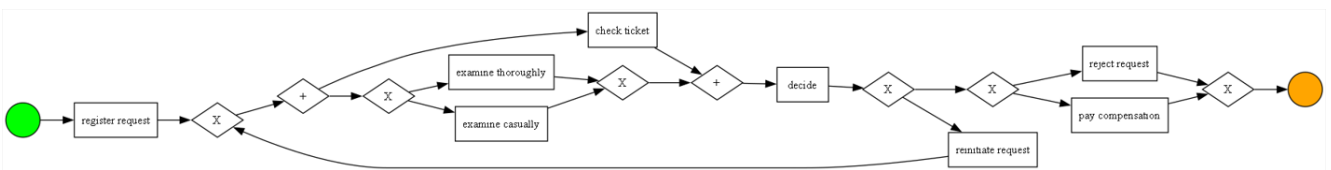
KUVA 8 Process mining työkalu (Microsoft 2024)

Pm4py (Process Mining for Python) on avoimen lähdekoodin kirjasto, joka on suunniteltu erityisesti prosessilouhintaa varten. Se tarjoaa laajan valikoiman algoritmeja ja työkaluja prosessidatan analysointiin, mallintamiseen ja visualisointiin. Kirjasto on rakennettu Python-ohjelmointikielellä, mikä tekee siitä joustavan ja laajennettavan alustan erilaisiin tarpeisiin. Koska pm4py on avoimen lähdekoodin projekti, käyttäjät voivat muokata ja laajentaa sitä omien tarpeidensa mukaan. Tämä on erityisen hyödyllistä, jos organisaatiolla on spesifejä vaatimuksia, jotka eivät täyty valmiilla kaupallisilla työkaluilla. Kuvassa 9 esitetään yksi esimerkki prosessivaiheiden visualisoinnista ja sen vaiheistuksesta.



KUVA 9 Prosessin visualisointi ja vaiheiden tulkinta pm4py:llä

Pm4py tarjoaa pääsyn laajaan valikoimaan edistyneitä algoritmeja, kuten prosessimallien automaattinen synteesi, poikkeamien tunnistus ja suorituskykyanalyysi. Kuvassa 10 käsitellään prosessivirtausta ja sen vaiheistuksia, sekä prosessien välisiä suhteita. Tämä mahdollistaa syvällisen ja tarkan prosessianalyysin. Työkalun tehokas hyödyntäminen edellyttää Python-ohjelmointikielen ja datankäsittelyn tuntemusta. Tämä saattaa asettaa haasteita organisaatioille, joissa ei ole saatavilla teknistä osaamista tai resursseja kehitystyöhön.



KUVA 10 Esimerkki pm4py työkalulla tehdystä prosessin visualisoinnista

Käytännön työssä näiden kahden työkalun yhdistäminen osoittautui erittäin tehokkaaksi lähestymistavaksi. Microsoftin Process Mining -työkalua käytettiin ensisijaisesti datan nopeaan visualisointiin ja yleiskuvan saamiseen prosessien kulusta. Sen avulla pystyttiin tunnistamaan keskeiset prosessivaiheet, yleisimmät prosessipolut sekä ilmeisimmät pullonkaulat. Pm4py:tä hyödynnettiin syvällisempään analyysiin, jossa tarvittiin tarkempaa kontrollia ja räätälöityjä ratkaisuja. Esimerkiksi monimutkaisten poikkeamien tunnistaminen ja erilaisten prosessivariaatioiden vertailu onnistui tehokkaasti pm4py edistyneiden algoritmien avulla. Pythonin joustavuus mahdollisti myös datan esikäsittelyn ja integroinnin muiden analytiikkatyökalujen kanssa. Datan analysointi prosessilouhinnan näkökulmasta on monimutkainen tehtävä, joka vaatii sekä tehokkaita työkaluja että syvällistä osaamista. Microsoftin Process Mining -työkalu ja pm4py tarjoavat yhdessä kattavan ratkaisun, joka sopii sekä perusanalyysiin että syvällisempään tutkimukseen.

Datan analysoimiseen on olemassa laaja kirjo työkaluja ja ratkaisuja, jotka voivat vastata erilaisiin tarpeisiin ja resursseihin. Tässä työssä saadut kokemukset osoittavat, että oikeiden työkalujen valinta ja niiden tehokas hyödyntäminen ovat keskeisessä asemassa prosessilouhinnan onnistumisessa. Organisaatioiden tulisi arvioida omat tarpeensa, resurssinsa ja tavoitteensa huolellisesti valitessaan työkaluja datan analysointiin.

4 YHTEENVETO TULOKSISTA

Lopputuloksena voimme todeta, että prosessilouhinnan työkalujen hyödyntäminen tarjoaa merkittäviä etuja liiketoimintaprosessien analysoinnissa ja optimoinnissa. Käyttämällä näitä työkaluja pystymme tunnistamaan erilaisia prosessikokonaisuuksia organisaation sisällä sekä mittaamaan niiden tehokkuutta tarkasti ja objektiivisesti. Tämä mahdollistaa prosessien pulonkaulojen havaitsemisen, resurssien käytön optimoinnin ja prosessien sujuvoittamisen, mikä johtaa parempaan tuottavuuteen ja kilpailukykyyn.

Prosessilouhinnan mahdollistaminen Business Centralin prosessivirtojen mukaisesti vaatii kuitenkin jonkin verran kehitystyötä. Järjestelmän mukauttaminen siten, että se kerää ja tallentaa tarvittavan datan prosessilouhintaa varten, edellyttää teknisiä toimenpiteitä, kuten oman räätälöidyn taulun luomista ja tapahtumatilaaajien (event subscriberien) ohjelmoimista. Nämä toimenpiteet mahdollistavat sen, että kaikki prosessien kannalta olennaiset tapahtumat tallennetaan järjestelmällisesti ja johdonmukaisesti analyysia varten.

Huolimatta siitä, että kehitystyötä tarvitaan, ratkaisu voidaan toteuttaa geneerisenä. Tämä tarkoittaa, että kerran kehitettyä ratkaisua voidaan soveltaa useissa eri organisaatioissa ilman, että jokaisen yrityksen tarvitsee aloittaa alusta. Geneerinen ratkaisu on joustava ja skaalautuva, ja se voidaan mukauttaa eri yritysten erityistarpeisiin pienin muutoksin. Tämä tekee siitä kustannustehokkaan ja ajan säästöä tuottavan vaihtoehdon, erityisesti yrityksille, jotka haluavat ottaa käyttöön prosessilouhinnan ilman suuria investointeja räätälöityyn ohjelmistokehitykseen.

On kuitenkin tärkeää huomioida, että datan tulkinta ja analysointi on jokaisen yrityksen kohdalla tehtävä omana projektinaan. Vaikka tekninen ratkaisu mahdollistaa datan keräämisen ja tallentamisen yhtenäisellä tavalla, jokaisella organisaatiolla on omat erityispiirteensä, prosessinsa ja toimintatapansa. Tämä tarkoittaa, että dataa on analysoitava yrityskohtaisesti, jotta voidaan ottaa huomioon organisaation yksilölliset tarpeet, tavoitteet ja toimintaympäristö.

Datan analysointi vaatii syvällistä ymmärrystä yrityksen liiketoimintaprosesseista sekä kykyä tulkita dataa oikeassa kontekstissa. Tämä saattaa edellyttää asiantuntijoiden osallistumista analyysiin, henkilöstön koulutusta sekä mahdollisesti analyysityökalujen räätälöintiä vastaa-

maan yrityksen tarpeita. On myös tärkeää kommunikoida analyysin tulokset selkeästi organisaation sisällä, jotta löydökset voidaan hyödyntää tehokkaasti päätöksenteossa ja prosessien kehittämisessä.

Ratkaisu mahdollistaa kuitenkin tämän kaiken erittäin hyvin. Geneerisen ratkaisun avulla yritykset saavat käyttöönsä tehokkaat työkalut ja menetelmät prosessidatan keräämiseen ja esikäsittelyyn. Tämä luo vankan pohjan datan analysoinnille ja tulkinnalle. Lisäksi, koska ratkaisu on joustava, se voi mukautua yrityksen kasvaessa tai sen tarpeiden muuttuessa, mikä tekee siitä pitkäikäisen investoinnin.

Yhteenvetona voidaan sanoa, että vaikka prosessilouhinnan käyttöönotto Business Centralissa edellyttää aluksi kehitystyötä ja resursseja, sen tuomat hyödyt ovat moninkertaiset. Yritykset voivat parantaa prosessiensa tehokkuutta, lisätä läpinäkyvyyttä ja tehdä parempia, dataan perustuvia päätöksiä. Tämä johtaa lopulta parempaan suorituskykyyn, asiakastyytyväisyyteen ja kilpailuetuun markkinoilla. Ratkaisun geneerisyys ja joustavuus tekevät siitä erinomaisen valinnan organisaatioille, jotka haluavat hyödyntää prosessilouhinnan mahdollisuuksia täysimääräisesti.

Työn pohjalta voidaan edistää jatkotutkimusaiheita, jossa viitekehystä laajennetaan esimerkiksi muihin organisaatiossa oleviin järjestelmiin. Usein prosessivirrat lähtevät liikkeelle esimerkiksi CRM-järjestelmästä, joten näiden prosessivirtojen louhinta voisi hyödyttää yritysosorganisaatiota ymmärtämään kokonais kuvaa paremmin. Jatkotutkimusta voisi tehdä myös datan laadun vaikutuksesta prosessilouhinnan tuloksiin, jossa voisi kehittää menetelmiä datan laadun arvioimiseksi ja parantamiseksi ennen prosessilouhintaa.

LÄHTEET

- Aalst, Wil & Adriansyah, Arya & Medeiros, Ana & Arcieri, Franco & Baier, Thomas & Blickle, Tobias & R.P., Jagadeesh Chandra Bose & Brand, Peter & Brandtjen, Ronald & Buijs, Joos & Burattin, Andrea & Carmona, Josep & Castellanos, Malú & Claes, Jan & Cook, Jonathan & Costantini, Nicola & Curbera, Francisco & Damiani, Ernesto & de Leoni, Massimiliano & Wynn, Moe. 2011. Process Mining Manifesto. Lecture Notes in Business Information Processing. Saatavissa: https://www.researchgate.net/publication/221585990_Process_Mining_Manifesto Viitattu: 13.4.2024.
- Acampora, G. Vitiello, A. Di Stefano, B. van der Aalst, W. M. P. Günther, C. W. & Verbeek, H. M. W. "IEEE 1849TM: The XES Standard: The Second IEEE Standard Sponsored by IEEE Computational Intelligence Society," IEEE Computational Intelligence Magazine. 2017. Viitattu: 13.4.2024.
- Berti, Alessandro & van Zelst, Sebastiaan & Aalst, Wil. Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science, 2019. Saatavissa: https://www.researchgate.net/publication/333130393_Process_Mining_for_Python_PM4Py_Bridging_the_Gap_Between_Process-and_Data_Science Viitattu: 14.8.2024.
- Consoli, S. Recupero, D. ja Saisana, M. Data Science for Economics and Finance. Springer Nature: Cham Saatavissa: <https://publications.jrc.ec.europa.eu/repository/handle/JRC124606> Viitattu: 13.4.2024.
- Demiliani, S. & Tacconi, D. 2024. Mastering Microsoft Dynamics 365 Business Central: The Complete Guide for Designing and Integrating Advanced Business Central Solutions. Second edition. Birmingham, England: Packt Publishing. Saatavissa: <https://www.oreilly.com/library/view/mastering-microsoft-dynamics/9781837630646/> Viitattu 14.9.2024.
- Dumas, M. La Rosa, M. Mendling, J. Reijers, H. Business Process Management. Springer Nature: Berlin, 2018. Saatavissa: <https://link.springer.com/book/10.1007/978-3-662-56509-4> Viitattu: 13.7.2024.
- González López de Murillas, E. Reijers, H.A. & van der Aalst, W.M.P. Connecting databases with process mining: a meta model and toolset. 2019. Saatavissa: <https://doi.org/10.1007/s10270-018-0664-7> Viitattu: 20.5.2024.
- Günther, C., and van der Aalst, W. 2007. Fuzzy mining: Adaptive process simplification based on multi-perspective metrics. BPM 2007. Springer. Saatavissa: https://www.researchgate.net/publication/221586306_Fuzzy_Mining_-_Adaptive_Process_Simplification_Based_on_Multi-perspective_Metrics Viitattu: 14.4.2024.
- Jawad, Z.N., Balázs, V. 2024. Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review. Saatavissa: <https://doi.org/10.1186/s43088-023-00460-y> Viitattu 21.5.2024.
- Jans, M. Martijn van der Werf, J. Lybaert, N. Vanhoof, K. 2011. A business process mining application for internal transaction fraud mitigation. Expert Systems with Applications. Vol. 38

Saatavissa:

https://www.researchgate.net/publication/220217942_A_business_process_mining_application_for_internal_transaction_fraud_mitigation Viitattu: 15.8.2024.

Kapulin, D. V., Russkikh, P. A. and Moor, I. A. 2019. Integration capabilities of business process models and ERP-systems. Journal of Physics: Conference Series 1333 Saatavissa: https://www.researchgate.net/publication/337349550_Integration_capabilities_of_business_process_models_and_ERP-systems Viitattu 1.9.2024

Leontjeva, A. Conforti, R. Di Francescomarino, C. Dumas, M. Maggi, F. Complex Symbolic Sequence Encodings for Predictive Monitoring of Business Processes. 2015. Saatavissa: https://www.researchgate.net/publication/281446750_Complex_Symbolic_Sequence_Encodings_for_Predictive_Monitoring_of_Business_Processes

Montali, M. Senderovich, A. ja Weidlich, M. *Process Mining Workshops*. Springer Nature Switzerland, 2023. Saatavissa: <https://library.oapen.org/bitstream/20.500.12657/62417/1/978-3-031-27815-0.pdf> Viitattu 6.3.2024.

Reinkemeyer, L. 2020. Process Mining in Action. Springer Nature Cham. Saatavissa: <https://www.springerprofessional.de/process-mining-in-action/17799536>

Sonawane, S. B. and Patki, R. P. 2015. Process Mining by using Event Logs. International Journal of Computer Applications. Saatavissa: https://www.researchgate.net/publication/276128910_Process_Mining_by_using_Event_Logs

Van der Aalst, W. Carmona, J. 2022. Process Mining Handbook. Springer Nature Cham. Saatavissa: <https://link.springer.com/book/10.1007/978-3-031-08848-3>

Van der Aalst, W. Weijters, A. ja Maruster L. 2004. Workflow Mining: Discovering process models from event logs. Eindhoven. Saatavissa: https://www.researchgate.net/publication/3297301_Workflow_Mining_Discovering_Process_Models_from_Event_Logs Viitattu: 4.7.2024

Van der Aalst, W.M.P. 2011. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Nature Heidelberg. Saatavissa: https://www.researchgate.net/publication/220693003_Process_Mining_Discovery_Conformance_and_Enhancement_of_Business_Processes Viitattu: 20.6.2024

Van der Aalst, W. M. P. 2019. Everything You Always Wanted to Know About Petri Nets, but Were Afraid to Ask. Business Process Management. BPM 2019. Lecture Notes in Computer Science. Vol. 11675. Springer, Cham. Saatavissa: https://www.researchgate.net/publication/335361233_Everything_You_Always_Wanted_to_Know_About_Petri_Nets_but_Were_Afraid_to_Ask Viitattu: 13.4.2024

Van der Aalst, W. Weijters, A. ja Maruster, L. 2002. "Workflow Mining: Which processes can be rediscovered," in BETA Working Paper Series, Eindhoven. Saatavissa: <https://pure.tue.nl/ws/portalfiles/portal/1925517/558655.pdf> Viitattu: 14.5.2024

Vikash, Y. Dubey, A. Dubey, G., Singh, P. ja Suryani, E. *Process Mining Techniques for Pattern Recognition*. CRC Press, 2022. Saatavissa: [Process Mining Techniques for Pattern Recognition \(oreilly.com\)](https://oreilly.com/catalog/errata.csp?isbn=9781496028955) Viitattu 6.3.2024.

Wu, Yingdan & He, Yijun & Wang, & Wen, Lijie & Luo, Yu-Feng. 2019. A Business Process Analysis Methodology Based on Process Mining for Complaint Handling Service Processes. Saatavissa: https://www.researchgate.net/publication/341977388_A_Business_Process_Analysis_Methodology_Based_on_Process_Mining_for_Complaint_Handling_Service_Processes Viitattu: 15.3.2024

Microsoft. 2024. Create a custom telemetry event. Saatavissa: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-instrument-application-for-telemetry-app-insights> Viitattu: 14.9.2024.

Microsoft. 2024. Create a custom table. Saatavissa: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-table-object> Viitattu: 14.9.2024.

Microsoft. 2024. Create codeunit. Saatavissa: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-codeunit-object> Viitattu: 16.9.2024.