

Kodolányi János Egyetem

SZAKDOLGOZAT

NAGY RICHÁRD
ÜZEMMÉRNÖK-INFORMATIKUS
ALAPKÉPZÉSI SZAK

Budapest

2024

**Kodolányi János Egyetem
Informatika Tanszék**

**Robotikus folyamatautomatizáció implementálása
nagyvállalati környezetben**

Konzulens: Dr. Pitlik László

**Készítette: Nagy Richárd
ÜZEMMÉRNÖK-INFORMATIKUS
ALAPKÉPZÉSI SZAK**

Budapest

2024

Kivonat

A szakdolgozat bemutatja a robotikus folyamatautomatizáció (RPA, robotic process automation, azaz robotikus folyamatautomatizáció) fontosságát és alkalmazását a vállalati környezetben. Az RPA lehetővé teszi az idő felszabadítását a manuális folyamatok csökkentésével, így növelve az értékteremtő tevékenységekre fordítható erőforrásokat. A dolgozat hangsúlyozza, hogy az RPA segítségével csökkenthetők az IT költségek, miközben nem igényel speciális informatikai előképzettséget a felhasználoktól.

A dokumentum részletesen ismerteti az RPA alkalmazását példákon keresztül, mint például az email küldését csatolmány nélkül vagy az adatok feldolgozását és rendezését. Emellett kiemeli az RPA előnyeit más informatikai fejlesztési módszerekkel szemben, hangsúlyozva, hogy az RPA egy üzleti szempontból tervezett eszköz, amelyet az üzleti szféra is könnyen használhat és fejleszthet.

A dolgozatban részletesen bemutatott robotikus folyamatautomatizáció implementálási tanulmányok és kódok segítségével a szerző konkrét példákon keresztül mutatja be az RPA gyakorlati alkalmazását. A dokumentum végén a kulcsszavak segítségével összegzi a legfontosabb fogalmakat és területeket, amelyeket az RPA kapcsán érdemes figyelembe venni.

Összességében a szakdolgozat átfogó képet nyújt az RPA alkalmazásáról és előnyeiről a vállalati környezetben, valamint konkrét példákon keresztül segít megérteni ennek a technológiának a gyakorlati használatát és implementálását.

Kulcsszavak

robot, robotika, RPA, folyamatautomatizáció, folyamatfejlesztés, információbiztonság, audit, tesztelés, jogosultság, log

Title of document

Implementing robotic process automation in corporate environment

Abstract

The thesis presents the importance and application of Robotic Process Automation (RPA) in the corporate environment. RPA enables time savings by reducing manual processes, thereby increasing resources available for value-added activities. The thesis emphasizes that RPA can reduce IT costs without requiring special IT skills from users.

The document extensively discusses the application of RPA through examples such as sending emails without attachments or processing and organizing data. It also highlights the advantages of RPA over other IT development methods, emphasizing that RPA is a business-focused tool easily usable and developable by the business sector.

Through implementation studies and codes detailed in the thesis, the author demonstrates the practical application of RPA through concrete examples. At the end of the document, key terms summarize the most important concepts and areas worth considering in relation to RPA.

Overall, the thesis provides a comprehensive overview of the application and benefits of RPA in the corporate environment, and through specific examples, it helps understand the practical use and implementation of this technology.

Keywords

robot, robotics, RPA (Robotic Process Automation), process automation, process improvement, information security, audit, testing, authorization, log

1. Bevezetés

Ebben a fejezetben bemutatásra kerül a szakdolgozatom célja, a kapcsolódó feladatok, valamint saját motivációm a téma választása tekintetében. Továbbá bemutatom a célcsoportot, és ismertetem a szakdolgozat hasznosságát a célcsoport számára. Valamint a feladatokhoz kapcsolódóan meghatározom, az implementáció mely része képezi a szakdolgozat részét, és mely része nincs bemutatva a dolgozatban.

1.1. Köszönetnyilvánítás

Szeretném megköszönni nem csak a szakdolgozatírás kapcsán kapott iránymutatást és közös ötletet, hanem az elmúlt 3 évben kapott támogatást tanárainnak, Dr. Rikk Jánosnak és Dr. Pitlik Lászlónak, akik nélkül ez a dolgozat sosem készült volna el.

1.2. Célok

Szakdolgozatom elsődleges célja teljes spektrumban bemutatni a napjainkban népszerű RPA szoftverek nagyvállalti környezetben történő implementálását, egy fiktív üzleti folyamat automatizálása mentén, azzal az alapossággal, amire tapasztalatim alapján mind maguk a beszállítók, mind esetlegesen harmadik félként megjelenő tanácsadó cégek sem szánnak elég erőforrást és kevesebb fókuszot helyeznek egy-egy pilot project implementálása mellett.

2016 óta foglalkozom szoftver robotok készítésével, 2 nagyvállalati környezetben is része voltam a kezdeti implementációs csapatnak, valamint a fejlesztői és üzemeltető csoportnak. Rengeteg olyan hibát vétettünk, aminek túl későn szántunk megfelelő hangsúlyt, ezzel értékes erőforrásokat „égettünk”, mint az üzleti, mind az implementációs oldalon, ami hitelvesztéssel, keserű szájjal is járt, amit hosszútávú kemény munkával tudtunk csak helyreállítani.

Éppen ezért úgy érzem elmondhatom magamról, hogy azért, hogy tudom, hogyan nem-kell robotot implementálni, tapasztalatommal és tudásommal egy átfogó képet szeretnék adni és bemutatni egy teljes, end-to-end RPA implementációt a szakdolgozatomban.

1.3. Feladatok

Feladatom a szakdolgozatomban egy saját fejlesztésű robotikus folyamaton túl bemutatni, milyen hardware és software követelmények merülnek fel az implementáció alatt, mi is az a robot, mit tud, mire alkalmas, hogyan érdemes kezelni a jogosultságait, milyen rendszerekhez férhet hozzá, mi történik azzal a feladattal, amit robot nem tud elvégezni, kvázi választ adok arra a kérdésre, hogy ki takarít a robot után, ha egy feladatot nem, vagy rosszul végez el.

Rendszerimplementáció után pedig olyan kérdésekre szeretnék választ adni, miszerint milyen folyamatot érdemes robottal automatizálni, ezeket a folyamatokat miként érdemes sorrendbe rakni, milyen követelményspecifikációt igényel meg egy robotikus folyamat létrehozása, majd ezeket a folyamatokat hogyan lehet visszamérni, milyen metrikák lesznek a segítségünkre.

1.4. Motivációk

Mai rohanó világunkban szinte már csak a most van jelen. Most akarunk mindent. Mindent is. De ha nem is most, elvárjuk a világtól, egymástól, cégektől, bankoktól, hogy kéréseinket elsőre, jól és időre teljesítsék. Viszont az erőforrásaink nem mindig állnak rendelkezésre ehhez. Eltűnődünk, hogy bizonyos dolgokat miért kell csinálnunk, miért nem csinálhatunk más dolgokat, olyanokat, amiknek valós értéke van, és bizonyos dolgokat miért nem csinál meg helyettünk valaki más. Nagyvállalati környezetben is megjelennek ugyanezek a kérdések.

Ha ugyanolyan fontos az összes projekt, viszont ugyanolyan fontos az üzletvitel fenntartása, a feladatokat száma viszont csak nő, de az erőforrásaink ezek a növekedést nem tudják követni, mint a túlfeszített húr, egyszer csak szétrobban a minőség háromszöge, amibe a feladatainkat akarjuk helyezni. Ennek a háromszögnek az oldalai a költség-határidő-feladatok terjedeleme. Az RPA segítségével ezt a három oldalt aktívan meg tudjuk támogatni.

RPA fejlesztőként elsődlegesen olyan szoftver robotokat készítünk, amik munkában töltött napjainkat teszik könnyebbé és kiveszik a manuális, lélekölő, agyzsibbasztó, hozzáadott értéket aligha képviselő feladatokat a kezeinkből. Az RPA képes időt felszabadítani, hogy egy vállalat munkatársai értékteremtő folyamatokon tudjanak dolgozni azáltal, hogy csökkenti a kézzel végzett folyamatokat. Ezt a felszabadult időt, és ez által erőforrást pedig sokszínűen lehet felhasználni; túlóra szükség csökkentése, minőség emelése, átfutási idő csökkentése.

A fenti állítás azonban szinte bármilyen eszköz használatával elérhető. Képesek vagyunk írni egy Python scriptet, ami egy terminál emulátorba (pl IBM i Access Client Solutions) adatokat visz fel, vagy olvas ki. Írhatunk C# kódot arra, hogy ügyfélleveleket készítsünk PDF fájlként, és azt küldjük el az ügyfélnek egy levelező szoftver API hívásán keresztül. De akkor miért jó az RPA?

Mert az RPA nem követeli meg azt a fajta informatikai előképzettségeket, amit egy tradicionális IT fejlesztés megkíván. Mert az RPA egy üzlet számára tervezett eszköz, amit az IT ugyan támogat, de az üzlet saját maga fejleszt. Mert képes csökkenteni a magas IT költségeket, azáltal, hogy hídként összeköt olyan rendszereket, amit nem képesek egymással kommunikálni, mert

vagy legacy rendszerekről beszélünk, amiket már nem fejlesztenek, de még fenntartanak, vagy szintén azért, mert egy ügyféligényhez képest irreálisan magas lenne a költsége.

Az RPA számomra olyan, mint az Excel. Várom azt az utópisztikus világot, amikor ugyanannyira elfogadott eszköz lesz egy céges gépen, mint a Microsoft táblázatkezelője. Biztos vagyok benne, hogy nem mindenki képes saját magának robotot építeni, ugyanúgy ahogy nem mindenki képes összetett függvényeket, VBA makrókat írni, de a lehetőség adott (hacsak céges szinten nincs tiltva a makró biztonsági fenyegetettség miatt, de erről a gondolatról később).

1.5. Célcsoportok

Egyéni vállalkozótól kezdve, KKV-ken át a multinacionális nagyvállalatokig mindenkinek érdeke a költség és erőforrás megtakarítás. Bármilyen olyan vállalat, akinek szabályon alapján leírható folyamataik vannak az tekinthető célcsoportnak. Biztosítási és bankszektorban dolgozó vállalatoknak, akiknek komoly MNB-s törvényi megfelelési kötelezettségük van, és erősen szabályozott környezetben működnek különösen segítségükre lehet egy olyan eszköz, ami ugyanazt a folyamatot mindig ugyanúgy hajtja végre. Ugyanakkor olyan adminisztratív feladata egy egyéni vállalkozónak is lehet, amihez nem szükséges a kognitív döntéshozatal, hanem szintén egy papírlapra felírható szabályrendszerre támaszkodva képes ezt a feladatot ellátni egy robot. Ezáltal bármilyen olyan vállalkozó/vállalat potenciális célközönségnek tekinthető, amelyek szabályszerű döntésekre támaszkodó folyamatot üzemeltetnek emberi erőforrás segítségével.

Megemlíthetném a személyes felhasználást is, de csak azért, mert létrehoznék egy robotot, ami reggel 6:00—kor felköszönti az éppen születésnapjukat ünneplő Facebook ismerőseimet, nem feltétlenül kell elmerülniük a jogosultságmenedzsment, céges szabályzatok vagy törvényi megfelelés tengerben, ugyanakkor egy hibabiztos robot felépítése és futtatása révén ők is tekinthetők célcsoportnak.

1.6. Hasznosság

Akkor tekinthetjük a dolgot hasznosnak, ha az képes valamilyen szükségletet kielégíteni, olyan területen, ahol a célcsoportnak hiányérzetet van. Azzal a hipotézissel élve, hogy egy stabilan működő vállalatnak érett, stabil, szabályozott folyamatokra van szükségük, ezeket a folyamatokat forráskódba is lehetne ültetni, és az üzleti logikát egy robotra bízni. Amennyiben képesek vagyunk olyan folyamatokat kiválasztani és robotizálni, amik kellően magas ismétlésszámban végeznek a munkatársak, és amiket jelenleg manuálisan végeznek el, emberi erőforrást felhasználva erre, akkor pozitív megtérülést érünk el akár időben számolva, akár

költségben. Általánosságban kimondható, hogy a nagyvállalatok többségének évről-évre a költségcsökkentés az egyik legnagyobb kihívása, aminek eléréséhez hasznos lehet a robotok implementálása. Így akár ugyanazzal az erőforrással több, vagy új munkafolyamatokat tudnak elvégezni, vagy a meglévő munkafolyamatot tudják kevesebb erőforrással is elvégezni. A felszabadult erőforrást pedig átképzik vagy leépítik, akár csak azzal, hogy az elvándorolt munkaerőt nem töltik vissza.

Másik oldalról, ha nem erőforrás oldalról közelítjük meg a hasznosságot, akkor az emberi hibafaktort is képesek lehetünk csökkenteni, mert a robot nem fárad el, a robot nem dolgozik betegen, a robot nincs beszorítva a napi 8 munkaórába, nem stresszel, így a megtanított feladatot minden esetben ugyanúgy fogja elvégezni. A kulcs itt pedig a megtanított munka részletességében rejlik, ami szintén felkészíti egy vállalatot arra, hogy megvizsgálja folyamatait, és kattintás-szinten is ledokumentálja őket. Ez nem csak a robot fejlesztése közben hasznos, hanem egy új belépő esetén is, aki az utcáról belépve, ezeket a dokumentumokat használva, akár az első órájától ezáltal önállóan tud dolgozni.

Bővebben pedig azok a vállalatok számára lehet hasznos a dolgozatom, akik éppen most fontolgatják, esetleg kezdik meg fel és kiépíteni saját RPA kompetencia központjuk, akár belső erőforrásból, akár külső beszállító segítségével. Vizsgálatommal és meglátásaimmal remélem hasznos része lehetek egy-egy ilyen implementációs projectnek.

1.7. A dolgozat szerkezetéről

Publikáció alapstruktúráját a tanszéki ajánlás adja, https://miau.my-x.hu/digeco/2020/2020oszdigeco_tdk_publication_ures.docx (2024.04.04), valamint a Kodolányi János Egyetem által közölt szakdolgozat minta: https://www.kodolanyi.hu/konyvtar/images/tartalom/File/Honlapra/Feltoltes/szakdoli_minta.pdf.

Stilisztikai ajánlasként az alábbi állomány került figyelembevételre: <https://miau.my-x.hu/myx-free/index.php3?x=test1> (2024.04.04)

A terjedelmi korlát a konzulenssel folytatott konzultáció alapján lett meghatározva

A szakdolgozat megfelel a tudományos írás (academic writing skills) alapvető szabályainak (Forrás: https://unipub.lib.uniconvinius.hu/4649/1/JelKep_2018_4_Deli_Eszter_Retvari_Marton.pdf, DOI: 10.20520 / JEL - KEP . 2018.04. 63, Jel-kép KOMMUNIKÁCIÓ, KÖZVÉLEMÉNY, MÉDIA 2018/4. szám)

Egyéb formai szabályok:

- a kulcsszavak, kulcsgondolatok félkövér formázással szerepelnek az értelmezés elősegítése érdekében
- az idegen nyelvű kifejezések a hiányzó magyar megfelelő miatt, valamint a szaknyelv által használt angolul kerültek felhasználásra
- az idézőjel használata kettős célt rejt: dőltbetűvel használva az idézeteket jelölöm, dőltbetű nélkül pedig a robot paramétereit/változóit és meghívott folyamjait jelölik
- kódrészletek képernyőképként történő beillesztése esetén a képek kerettel rendelkeznek, balra igazítva, míg a többi, nem kódrészlet tartalmú képernyőképek középre igazítva, felirattal, keret nélkül, az ábrajegyzékben is feltüntetve szerepelnek

Feladatok rövid bemutatása

- A következő feladatok képezik részét a szakdolgozatnak:
- Követelményspecifikáció megírása és bemutatása
- Megvalósítási tanulmány megírása és bemutatása
- Felhasználó, valamint jogosultság igénylés
- RPA könyvtárak elkészítése (email, adattárház)
- Robot kód fejlesztése
- Queue-k létrehozása
- Asset-ek létrehozása
- Adatvizualizáció, riportok, visszamérés
- Robot kód üzleti tesztelése
- Robot élesítése
- Robot üzemeltetése

A következő lépések pedig csak érintve képezik részét a szakdolgozatnak, beleértve az indoklással:

- **Előzetes IT infrastruktúra (hardware) kialakítása:** kiindulópontnak azt vettem, hogy a robot fejlesztői és éles környezete már ki van építve. Ez azt jelenti, hogy létre lett hozva a virtuális környezet, ahol a robotok fejlesztése, tesztelése és üzemeltetése történik. Minden üzemeltetésre használt virtuális gépre telepítve lett a UiPath robot és Studio komponens. Valamint ki lett alakítva a robotok számára egy munkaköri sablon, ami rögzíti azokat a jogosultságokat, amik minden robot számára szükségesek, például hálózati meghajtó jogok. Ugyancsak telepítve lettek az általános applikációk és

rendszerek, amik a munkavállalók számára is elérhetőek (vagy szoftverközpontból letölthetővé tették).

- **Windows munkakörnyezet (software) kialakítása:** a robot számára létrehozott környezet teljesen azonos egy munkavállaló számára létrehozott környezettel. Csak említtem mely rendszerek telepítése szükséges a virtuális gépre.
- **Szervezeti megoldások (orgware)**
 - **Új felhasználó létrehozása:** ez egy általános helpdesk feladat, kvázi új belépő rögzítése az IBM Security Identity Managerben.
 - **Jogosultságok megadása:** szintén általános helpdesk feladat, csak említés szintjén érintem, nem mutatom be a teljes munkafolyamatot.

Lejjebb pedig a szaknyelvben használt fogalmakat, rövidítéseket szeretném bevezetni, amit a szakdolgozat további részében használni fogok

- **UiPath:** a RPA megoldások fejlesztéséhez és üzemeltetéséhez használt szoftver neve
- **Orchestrator:** az Orchestrator teszi lehetővé az RPA folyamatok központi kezelését, ellenőrzését és ütemezését egyetlen felhasználói felületen keresztül
- **Studio:** A (UiPath) Studio egy teljes körű fejlesztői környezet, amely lehetővé teszi a robotikus folyamatok létrehozását, szerkesztését és kezelését
- **Queue:** az Orchestrator egyik egysége, olyan tároló, amiben a feldolgozásra előkészített, vagy már feldolgozott tranzakciókat, és azok paramétereit tárolhatjuk
- **Queue Item:** egy adatelem, amely bármilyen queue-ban található tranzakciót és annak paramétereit jelenti
- **Asset:** az asset egy olyan érték vagy beállítás, amelyet az Orchestrator-ban tudunk tárolni, dinamikusan módosítani és más folyamatok számára elérhetővé tenni. Az asset-ek lehetnek például URL-ek, fix értékek, email disztribúciós listák vagy más konfigurációs adatok. A credential típusú asset az asset-ek egy különleges fajtája, ez egy felhasználónév és jelszó párost képes tárolni, ahol a jelszó titkosított. Az asset-ek létrehozása lehetővé teszi, hogy ezeket a központi helyről könnyen kezeljék az üzemeltetők és elérjék a robotok.
- **Selector:** a selector a UiPath Studio-ban a robotok által használt egyedi azonosító. Egy selector egy XML-formátumú kifejezés, amely az adott alkalmazás vagy weboldal elemére utal.

- **Connector:** olyan kód, illetve kódok összessége, ami egy adott másik program leggyakrabban használt funkcióit csomagosítja, az újrafelhasználhatóság és üzemeltetés érdekében
- **Dispatcher:** olyan tabuláris adatokkal dolgozó keretrendszer, ami adott strukturált adatból (táblázat, adattábla, mappában fájlnevek listája. stb) queue item-(ek)et hoz létre.
- **Performer:** az robot kód, ami a queue-ba olvasott adatokat ténylegesen feldolgozza az előre definiált szabályok szerint.
- **REFramework** (Robotic Enterprise framework): state-machineből álló keretrendszer. A használt state-eket (Init, Get Transaction Data, Process és End), valamint a köztük lévő átmeneteket írja le.

2. Szakirodalmi háttér

Ebben a fejezetben téma szerint csoportosítva, alfejezetekre bontva bemutatom a felhasznált szakirodalmat, valamint a szakdolgozati téma és a képzés tantárgyainak kapcsolatát.

2.1. Gazdasági, vállalatirányítási szempontból

Kezdjük azzal, mi is az az RPA, vagyis robotikus folyamat automatizáció. *„Robotic process automation (RPA) is a software technology that makes it easy to build, deploy, and manage software robots that emulate humans actions interacting with digital systems and software. Just like people, software robots can do things like understand what’s on a screen, complete the right keystrokes, navigate systems, identify and extract data, and perform a wide range of defined actions. But software robots can do it faster and more consistently than people, without the need to get up and stretch or take a coffee break.”* (Forrás: <https://www.uipath.com/rpa/robotic-process-automation>, 2024.03.26). Itt fontos megjegyezni, hogy mire gondolhatunk, az *easy*, vagyis a megoldás *könnyű* készítése, élesítése és menedzselése esetén? Véleményem és tapasztalatom szerint ez a már kiépített robot infrastruktúra, és fejlesztési folyamat esetén úgy értendő, hogy költség és folyamat szempontból vizsgálva olcsóbb, valamint gyorsabb, mint egy tradicionális IT fejlesztés. Az *understand* szó pedig szintén magyarázatot érdemel. Itt a robot nem a képernyőn lévő adatot képes megérteni, mint adatvagyon, vagy mint összetett gondolat, hanem a képernyőn lévő GUI (graphical user interface, grafikus felhasználói felület) adatait képes érzékelni (például egy Google Chrome ablak meghatározott aaname értékű selectora).

A fenti állítás érdekessé teheti a robotizáció bevezetését, mind a KKV-k, mind a nagyvállalatok számára, de mégis milyen folyamat esetében érdemes ezt a technológiát választani? *„Jellemzően olyan feladatoknak a robotizálása célszerű, amelyek gyakran, standard módon zajlanak le, előre rögzített szabályok szerint, nem túl komplexek, s nem a vállalat, intézmény szempontjából a legkritikusabbak. Ezek jellemzően azok a feladatok, amelyeket el kell végezni, de végrehajtásuk nem okoz különösebb motivációs többletet a munkatársak számára.*

Példa: tipikusan ilyen robotizálható tevékenység egy emailben kapott információ rögzítése egy ügyviteli rendszerbe (vagy egy Excelbe), a csatoltan kapott file lementése megfelelő mappába, adott névkonvenció szerint.” (Forrás: <https://www.controllingportal.hu/amit-az-rpa-rol-tudni-erdemes/>, Bitó János, 2018.03.20.) Itt is pontosítanunk szükséges a *gyakran*, a *nem túl komplex* és *legkritikusabb* folyamatokat, mert ezeket többféle módon értelmezhetjük. A gyakran önmagában értelmezhetetlen. Dolgoztam már olyan roboton, amit évente egyszer kell csak

használni, de akkor 2 hétig folyamatosan az üzleti folyamán milyensége miatt, mégse kizáró ok az automatizálásra. Itt én az frekvencia és volumen szorzatát használom, mint irányadó érték. A *nem túl komplex* jelző is értelmezhetetlen egymagában; ezt a megtérüléssel együttesen használom, mint erőfeszítés-hatás (benefit-effort) mátrixot. A *legkritikusabb* szó pedig egy teljesen felesleges jelző; szintén automatizáltam már olyan folyamatot, ami az idő függvényében volt kritikus (pl 1 órás szolgáltatási szint megállapodással rendelkező tranzakció, ami az idő tekintetében volt a legkritikusabb folyamata egy osztálynak).

Milyen előnyökkel jár ez egy vállalat számára? „*A robotizált ügyvitel előnyei többek között a minőség javításában (például az emberi hibák elkerülése vagy számosságának csökkentése), illetve a költségek csökkentésében érhetők tetten. Az RPA használata által csökkenhet a szándékos vagy véletlen emberi hibák lehetősége, rövidülnek az átfutási idők, illetve a monoton és jellemzően unalmas feladatok robotizálásával javulhat a munkavállalói elkötelezettség.*” (Forrás: Musinszki Zoltán, A digitalizáció hatása a back office munkára, In: Veresné, Somosi Mariann; Sikos, T. Tamás (szerk.) A fenntarthatóság holisztikus megközelítésben, Budapest, Magyarország: Akadémiai Kiadó (2023) 501 p. pp. 107-122., 16 p.). Ezzel a gondolattal teljes mértékben egyetértek, sőt, különösen fontosnak tartom egy-egy robotikus folyamat eredményének visszamérését a megadott KPI(-ok) (key performance indicator, kulcsfontosságú teljesítménymutató(k)) alapján, hogy teljesíti-e az elvárásokat, adott esetben alul-, vagy túlteljesíti-e őket.

Az RPA felhasználási területe széleskörű, ugyanakkor vannak egyértelmű iránymutatások, milyen területekkel érdemes kezdeni. „*Az RPA-t informatikai támogató és fejlesztő feladatokra alkalmazzák a leggyakrabban. Ezután következnek a pénzügyi-számviteli, a marketing és az ügyfélszolgálatos feladatok a Statista kutató-elemző cég adatai szerint.*” (Forrás: https://hu.wikipedia.org/wiki/Robotiz%C3%A1lt_folyamatautomatiz%C3%A1l%C3%A1s., Wikipedia, 2024.03.26.). Véleményem szerint a marketing egy széleskörű terület, itt mindenképp kiemelném, hogy a marketing melyik területén javasolt a robotok használata, leginkább a marketing adatok elemzésében és riportálásában érzem az RPA lehetőségét.

2.2. IT szempontból

Milyen további előnye lehet? Az RPA képes demokratizálni a szoftverfejlesztést, és az irányítást az üzlet kezébe adni. „*Another motivation for RPA was the chronic overload of programming departments. The programmers had too many orders. Too much optimization potential remained unconsidered and the end users were disappointed. Some automation*

approaches in the ERP and Office environment were realized by dedicated business experts, this was called shadow IT. With RPA this bottleneck can be bypassed. Automations that are within the realization horizon of business experts no longer requires a technical programmer. More colleagues can now automate their own processes.” (Forrás: <https://www.linkedin.com/pulse/rpa-lead-us-automation-duality-stefan-schnell>, Stefan Schnell, LinkedIn, 2020.11.30)

Ez viszont nem jelenti azt, hogy bárki, bárhogyan, bármikor robotot elkészíthet. Fontos, hogy a lenti alapelveket tartsuk szem előtt:

- „1. Kezdjük kis lépésekkel, de tervezzünk hosszú távra*
- 2. Építsünk szakértői csapatot, amely ismeri az RPA működés minden aspektusát*
- 3. Edukáljuk a kollégákat, hogy csökkentsük a robotizációhoz kapcsolódó félelmeket, és teret adjunk az ötleteknek*
- 4. Vegyük figyelembe a szabályozási környezet elvárásait*
- 5. Törekedjünk stabil, robusztus, ezáltal biztonságos megoldások kialakítására”* (Forrás: <https://fintechzone.hu/szoftverrobotok-rpa-segitik-penzugyeink-intezeset-is/>, Urbán Anikó, 2023.10.09.)

Az RPA ugyanakkor nem csak meglévő emberi munka utánzására alkalmas, tökéletesen alkalmas arra, hogy más, diszruptív technológiákkal lehet ötvözni, mint például az intelligens karakterfelismerés: *„Az intelligens dokumentum-feldolgozási megoldások képesek feldolgozni ezeket a dokumentumokat, azonban gyakran speciális összekötőket vagy RPA-robotokat használnak különböző forrásokból érkező dokumentumok gyűjtésére, mint például e-mail mellékletekből, mappák vizsgálatából, FTP-kiszolgálókról és más helyekről.... A kinyert adatok strukturált formátumba vannak rendezve, amely könnyen elérhető és felhasználható más alkalmazások vagy robotok számára az automatizált feldolgozáshoz.”* (Forrás: <https://goodroid.hu/intelligens-dokumentumfeldolgozas/>, 2024.03.26.), vagy a chatbot: *„Customers hate waiting in a queue to be served. This is as true for online activities as it is for activities in the physical world. This is where chatbots shine. Many questions that customer service reps deal with are easy to solve. Rather than imposing tiresome touch-tone menus on customers or asking them to listen to supposedly soothing music as they wait for a human to help them, many businesses are turning to chatbots. Chatbots can understand what customers are saying (or typing) through natural language processing (NLP) technology and can turn to RPA software robots (“bots”) to retrieve the relevant information from the appropriate*

systems. If any challenge arises, the customer can be routed to a human worker.”. (Forrás: <https://www.automationanywhere.com/company/blog/rpa-thought-leadership/why-combine-chatbots-and-rpa>, 2024.03.26).

2.3. HR szempontból

A robotok esetében is felmerül ugyan azon félelem, mint az első, illetve második ipari forradalom esetén, hogy a gépek elveszik a munkánk, ami koránt sincs így. *„A történelemben már többször előfordult, hogy drasztikusan megváltoztak a munkakörülmények – elég csak az ipari forradalmakra gondolni. A 18. század végén az első ipari forradalomban megjelentek a gőzgépek, majd rá száz évre a másodikban az elektromosság és a tömegtermelés. Az azóta eltelt 50 évben a 3. és 4. ipari forradalom is lezajlott, teret nyitva az automatizációnak és intelligens rendszereknek. Egyértelmű, hogy a fejlődés üteme felgyorsult, és nehéz megmondani, hogy mit hoz a jövő, ugyanakkor az elmondható, hogy a fejlődés eddig általában a munkavállalók javát is szolgálta. Korábbi foglalkozások eltűntek és újak jelentek meg, javítva az emberek életkörülményein.”* (Forrás: <https://mnbintezet.hu/tenyleg-elveszik-a-gepek-az-emberek-munkajat/>, Kotymán Alex, 2023.03.20). Ez felveti azt az utópisztikus kérdést, hogy mit történik, ha a mesterséges intelligencia odáig fejlődik, hogy ténylegesen minden munkát képes elvégezni? Képes-e elérni egy olyan állapotba az emberiség, hogy már nincs szükség emberre a feladataink elvégzéséhez, ezáltal megteremtve egy egyfajta alanyi alapjövedelemet a robotok által elvégzett munka béréként? A következő évtizedekben remélem erre is választ kapunk.

2.4. Üzleti szempontból

Ugyanakkor nem csak a munkavállalók életében okoz ez változást, az üzleti elemzők számára is új készségeket, új látásmódot igényel az RPA. *„Most new RPA managers think that the BAs they need for RPA are the same BAs they have within the IT Departments. That cannot be further from the truth. Multiple times, I needed to train 'traditional' Business Analysts to be able to perform their tasks within RPA. These BAs always assured me that they never needed to collect information in such granularity as they had to within their new roles. ... We need structured data, crystal clear decision points, process steps that are cleared of the dust time accumulated on them. It might look like a whole different process than what have been done before.”* (Forrás: <https://www.linkedin.com/pulse/why-writing-process-definition-document-rpa-easy-most-papp/>, Papp Bálint László, LinkedIn, 2021.03.28). Ezzel az új kihívással teljes mértékben egyetértek, hiszen akár csak egy olyan elágazás vagy lehetőség, ami nem kerül definiálásra, a szabályrendszert követő robotunk már nem képes a feladatának elvégzésére,

rosszabb esetben a tételek hibás feldolgozását eredményezi úgy, hogy ezt a robot észre sem veszi, és visszajelzést sem képes emiatt adni róla.

Azért is fontos, hogy az üzleti elemzők képesek legyen ezen új fajta követelmények dokumentálására, mert robotok esetében a folyamat lépéseit automatizáljuk, pontosan úgy, ahogy azt az emberek is végrehajtják. Ezen a ponton pedig érdemes a folyamatot magát is megvizsgálni, mert egy rossz folyamat automatizációja képes negatív megtérülést eredményezni. *„Bad processes create inefficiency and require extra effort to handle work-around procedures, rework cycles and duplicate tasks. They also increase the time and energy spent escalating issues to more expensive humans.”* Ugyanakkor ennek a megközelítésnek is lehet pozitív kimenetele, ha megvizsgáljuk a „zseblámpa effektust”. *„The probability of fixing a bad process is much higher when you shine a light on it by automating it.”* (Forrás: <https://isg-one.com/articles/why-automate-a-bad-process>, 2024.03.26.). Itt is fontos megjegyezni, mit értünk rossz folyamat alatt. A rossz jelző itt az a különbség, ami az erőforrások felhasználásának jelen állapota és a leghatékonyabb állapota közötti különbség számszerűsítve, ahol a jó, rossz, közepes szavakat bármilyen szabály szerint, de hozzá tudjuk rendelni ehhez a skálához.

Ezt a gondolatot folytatva, nem túlzás azt mondani, hogy folyamat automatizáció előtt, az üzleti folyamatot érdemes Lean módszertan szerint elemezni. *„...az általános tapasztalat szerint, ha a robotizálást hatékony, azaz lean folyamatra illesztjük, akkor a hatékonyság nő, ha pedig kevésbé hatékony folyamat esetében alkalmazzuk, akkor csak felerősödnek a hibák, gyakoriabbak lesznek és még több manuális munkát okoznak. Robotizálás előtt tehát mindenképpen érdemes átnéznünk és fejlesztenünk a folyamatainkat!”* (Forrás: <https://lean.org.hu/eszkozok/hogyan-kerulnek-robotok-az-irodaba/>, Nagy Richárd, 2021).

Egyes üzleti folyamatok szükségessé teszik a robotok szakértői rendszerként való működéseket. A bemenő adatok alapján szükséges meghatározniuk milyen döntést hozzanak meg, vagy milyen adatot állítsanak elő. (vö: <https://miau.myx.hu/miau2009/index.php3?x=e0&string=szak%C3%A9rt%C5%91i%20rendszer>).

Az alábbi kép és csatolmányok egy párhuzamosan futó, másik project eredményeképpen születtek, viszont a szakdolgozatnak nem képezi szerves részét.

Státusz	Company ID	Service ok1	SNOW duplikát ellenőrzés	OBT user	SNOW adat viz	Fedezet vizsgálat	M-site user
CANCELLED Riportba: Törölt ellátás + SNOW jegyzet kiírás							
OPEN vagy üres	ÜRES Riportba: Azonosítatlan ügyfél						
	NEM ÜRES	CHANGE	DUPLICATE Riportba: Duplikáció				
			NOT DUPLICATE	ÜRES			
				NINCS ÜGY	mind a négy OK SNOW Support + Szervezési feladat ELSE SNOW Support + Fedezetellenőrzési feladat		
				Service	Riportba: Ehhez a M-SITE Supporthez létezik már Support/Service		
				Support	Riportba: Ehhez a M-SITE Supporthez létezik már Support/Service		
				CLIENT	mind a négy OK SNOW Service update + Email küldés + Katalógus Service update		
				Service	ELSE	CLIENT SNOW Service update+ SOS felülvizsgálati feladat+ Email küldés + Katalógus Service update OPERATOR SNOW Service update+ Email küldés + Katalógus Service update + Riportba: Felülvizsgáló operátori tevékenység	
				Support	mind a négy OK SNOW Supportból Servicenytás + Email küldés + Katalógus Service update		
				ELSE	CLIENT SNOW Supportból Servicenytás + SOS felülvizsgálati feladat + Email küldés + Katalógus Service update OPERATOR SNOW Supportból Servicenytás + Email küldés + Katalógus Service update + Riportba: Felülvizsgáló operátori tevékenység		
				NINCS ÜGY	mind a négy OK SNOW Servicenytás + Email küldés + Katalógus Service update		
				ELSE	CLIENT SNOW Servicenytás+ SOS felülvizsgálati feladat+ Email küldés + Katalógus Service update OPERATOR SNOW Servicenytás+ Email küldés + Katalógus Service update + Riportba: Felülvizsgáló operátori tevékenység		
				OPERATOR	mind a négy OK SNOW Service update + Email küldés + Katalógus Service update		
				Service			

1. ábra Robot, mint szakértői rendszer (Forrás Kiértékelési_fa_szakértői_rendszer.xlsx és Szakértői_rendszer.xaml
mellékletek - saját közlés)

2.5. A mesterséges intelligencia szempontjából

És merre tart a jövő? Mikor fogja legyőzni az embereket a gép? Tekinthejtük-e intelligenciának azt a fajta leíró tudást, amivel egy általános, modern irodai munkavállaló rendelkezik, aki pontosan tudja a standardizált, szabályrendszerre épülő, strukturált adatokkal működő üzleti folyamatok egymásra épülő lépéseit? Ez alapján átültethető-e ez a tudás egy robotba, ami így üzleti intelligenciává válik? *“Több mint húsz éve, hogy az IBM számítógépe, a Deep Blue megverte az aktuális sakkvilágbajnokot, Garri Kaszparovot. Ma már a mobiltelefonon futó sakkalkalmazások is könnyedén le tudnak győzni nagymestereket. Alig húsz évvel a Deep Blue diadala után, 2016-ban a Google AlphaGo alkalmazása megverte I Szedol dél-koreai gónagymestert, akit a megelőző öt évben senkinek nem sikerült. (Szedol fel is hagyott profi gópályafutásával.) Deep Blue még irgalmatlan géperejére támaszkodott: titka az óriási memóriában rejtett, amely képessé tette arra, hogy minden lépés után lehetséges helyzetek millióit értékelje ki. Az AlphaGo ennél is többet tudott: a neurális hálózatok és a gépi tanulás módszerével gondolkodni kezdett, és olyan játékban nyert könnyedén, amelyet mi, büszke emberek az intuíció és a kreativitás bevehetetlen terepének tartottunk. Az azóta eltelt négy év pedig ennek a „nagymesternek” a sokszoros meghaladásáról szólt: az AlphaGo „utódai”, az AlphaGo Zero és az AlphaZero már az embert tönkreverő ősök felett arattak elsőprő*

győzelmeket, alig néhány napnyi gépi tanulás után. A 2019 végén bemutatott MuZero pedig már úgy diadalmaskodott összes elődje felett, hogy előzetesen még a játékszabályokat sem tanították meg neki – mindenfajta készen kapott tudás nélkül képes kiszámítani a győzelemhez vezető lépéseket. Rég túl vagyunk tehát azon a kérdésen, hogy a gép felülmúlhatja-e az embert – a gépek évente múlják felül újra és újra az embert felülmúló gépeket.” (Forrás: Boros Tamás - Filippov Gábor (szerk.): Magyarország 2030. Jövőkép a magyaroknak (Budapest, 2020))

Ha pedig a ChatGPT-t, egy generatív mesterséges intelligenciával rendelkező robotot kérdezzük a szoftverrobotok helyzetéről, hogy miként képes az RPA segíteni napjaink vállalatainak egyre fontosabbá váló feladatában, az információbiztonság fenntartásában, az alábbi választ kapjuk:

„A robotikus folyamatautomatizáció (RPA) képes jelentős mértékben hozzájárulni az információbiztonsághoz az alábbi módon:

- **Konzisztens végrehajtás:** Az RPA segítségével végrehajtott folyamatokat rendkívül konzisztensen lehet végrehajtani, minimalizálva ezzel az emberi hibák lehetőségét. Ez csökkenti az adatvesztés, adathalászat és más biztonsági fenyegetések kockázatát.
- **Felhasználói hozzáférés kezelése:** Az RPA alkalmazások szigorúan kezelik a felhasználói hozzáféréseket és jogosultságokat. Csak az engedélyezett felhasználók érhetnek hozzá az adatokhoz és végrehajthatnak bizonyos műveleteket, csökkentve ezzel a belső fenyegetéseket.
- **Auditálhatóság:** Az RPA rendszerek által végzett műveletek nyomon követhetők és auditálhatók. Ez lehetővé teszi az esetleges biztonsági incidensek gyors azonosítását és megfelelő intézkedések meghozatalát.
- **Adatkezelés és adatvédelem:** Az RPA rendszerek szigorúan betartják az adatvédelmi szabályozásokat és a compliance előírásokat. Az adatok manipulációja, tárolása és átvitele biztonságos környezetben történik, minimalizálva az adatsértések és adatlopások kockázatát.
- **Kiberbiztonsági automatizáció:** Az RPA integrálható különböző kiberbiztonsági rendszerekkel, így például tűzfalakkal, behatolásérzékelő rendszerekkel vagy malware-detektáló szoftverekkel. Ez lehetővé teszi a gyorsabb reakciót a biztonsági incidensekre és az automatizált védelmi intézkedések végrehajtását.
- **Automatizált incidenskezelés:** Az RPA alkalmazások segítségével az incidensek kezelése és reagálása gyorsabbá és hatékonyabbá válik. Az automatizált rendszerek

képesek azonnal érzékelni és reagálni a biztonsági incidensekre, minimalizálva ezzel azok hatását és csökkentve a válaszidőt.

Összességében az RPA segítségével a vállalatok jobban felkészülhetnek és reagálhatnak az információbiztonsági kihívásokra, miközben javítják az operatív hatékonyságot és csökkentik a humán hibák kockázatát.” (Forrás: ChatGPT, <https://chat.openai.com/c/4e163902-4986-448b-b174-b5d62177f230>, 2024.03.28)

2.6. A probléma/jelenség története

A szakdolgozatban az alábbi üzleti igényt fogom kiszolgálni, természetesen anonimizálva, ami egy olyan problémára keres megoldást, ami egy rugalmas kimutatáskészítő keretrendszer szeretne készíttetni, ami egy vállalati adattárházban automatikusan futtatott lekérdezések képes kimenteni hálózati meghajtó(k)ra és e-mailben való kiküldésre, Excel (xlsx) és csv formátumban.

Az igényt természetesen több módon, eszközzel is ki lehet(ett volna) szolgálni, de a mi esetünkben nem volt IT fejlesztési kapacitás, mert az igény alacsony üzleti értéket képviselt, és emiatt alacsony prioritást kapott.

Fejlesztés mindenképpen szükséges lett volna, vagy új rendszerimplementáció (ami szintén IT erőforrást használ), de ennek költsége messze magasabb lett volna, mint a folyamattal töltött emberi, manuális munkavégzés bérköltséggé alakított összege a következő 2 évre tekintve. Emiatt az alacsony ROI (return on investment, azaz a befektetésarányos megtérülés) alkalmazzunk olyan taktikai megoldást, mint a robot.

Az RPA a dolgozatomban, mint a tradicionális IT fejlesztések alternatívájaként van bemutatva. Amit képesek vagyunk RPA-val automatizálni, az más eszközzel is automatizálhatjuk. Hogy milyen különbség lehet egy RPA automatizáció és egy JavaScript kód között, azt a következő gondolat kísérlet alapján mutatom be, és teremtem meg a fejlesztési alternatívát és benchmarkingot. A gondolat kísérletben érintett feladat pedig egy olyan kód létrehozása, ami egy adott wikipedia oldal (https://en.wikipedia.org/w/index.php?title=Robotic_process_automation&action=history, 2024.04.10) változtatásait hasonlítja össze.

Compare selected revisions	
☐ (cur prev)	☒ 13.13, 29 February 2024 MrOllie (talk contribs) ... (24,769 bytes) (~265) ... (Reverted 1 edit by Code-Stew (talk): Also spam) (Tags: Twinkle, Undo)
☐ (cur prev)	☐ 4340, 29 February 2024 Code-Stew (talk contribs) ... (25,034 bytes) (+265) ... (Added " Web Automation " with reference link) (Tags: Manual revert, Reverted, Visual edit)
☐ (cur prev)	☒ 12.19, 29 February 2024 MrOllie (talk contribs) ... (24,769 bytes) (~265) ... (Reverted 1 edit by Code-Stew (talk): Blog spam) (Tags: Twinkle, Undo, Reverted)
☐ (cur prev)	☐ 4400, 29 February 2024 Code-Stew (talk contribs) ... (25,034 bytes) (+265) ... (RPA is used for web automation and it wasn't specified so I added it.) (Tags: Reverted, Visual edit)
☐ (cur prev)	☐ 13.56, 18 January 2024 Jacona (talk contribs) m ... (24,769 bytes) (+4) ... (Reverted edits by 103.179.211.194 (talk) (HG) (3.4.12)) (undo) (Tags: Huggle, Rollback)
☐ (cur prev)	☐ 13.56, 18 January 2024 103.179.211.194 (talk) ... (24,765 bytes) (~4) ... (undo) (Tag: Reverted)

2. ábra Wikipedia história összehasonlítás robottal és JavaScript-tel

Robottal az fenti képernyőképen lévő pirossal kiemelt rádiógomb leíró selectora az alábbi:

```
<html app='chrome.exe' title='Robotic process automation: Revision history - Wikipedia' />
<webctrl id='mw-oldid-1196793217' tag='INPUT' />
```

Ugyan ez az elem a weboldal html kódjában így néz ki:

```
<li data-mw-revid="1196793217" class="mw-tag-huggle mw-tag-mw-rollback"><span class="mw-history-histlinks mw-changeslist-links"><span>
```

A logika, ami alapján a párokat kiválasztjuk az összehasonlításhoz azonos, pusztán a megvalósítás különböző.

2.7. A szakdolgozati téma és a szak tantárgyainak kapcsolata

- Az **emberi viselkedés és kommunikáció**, az **európai civilizáció és identitás**, valamint a **komplex társadalomtudományi ismeretek** tantárgyakat érintem a szakirodalmi háttér fejezetben, ahol a 4 ipari forradalom találmányait és mai modern életünkre okozott hatásait mutattam be
- A **vállalati gazdaságtan** a szakdolgozat azon részeként van kiemelve, miszerint egy vállalatnak a versenyképessége megőrzése érdekében miként tud operációs költséget csökkenteni, a minőség fenntartásával (akár emelésével). Ahogy korábban is említettem, a költségek csökkentése, vagy emelkedésének féken tartása kruciális a vállalatok életében, főleg, ha az elmúlt évek inflációs adatait vizsgáljuk. Üzleti folyamatink hatásos automatizációjával ezek a költségek csökkenthetők, vagy éppen a költségek átcsoportosíthatók. Ha a bértömeg állandó marad, és a szükséges munkaerőt tudjuk képesek vagyunk csökkenteni az elvándorlás menedzselésével, a felszabadított költséget használhatjuk infláció feletti fizetésemelés allokálásra.
- A **vezetési és vállalkozási ismeretek** tantárgy keretében tanultak is fontos részét képezik egy RPA implementációnak. Mint jó vezető, mire kell figyelni a munkavállalót érintő változás esetében? Hogyan kezeljük azt a valós vagy vélt félelmét, hogy a robotok „elveszik a munkáját”. Miként készülhet fel egy modern kori vezető a modern kori géptörővel? Nagy hangsúlyt kell helyezni a változásmenedzsmentre, a nyílt, tiszta, érthető és átlátható kommunikációra, hogy egy vállalatnak milyen céljai vannak a

robotizációval. Mint vezető, mindig is előrébb jár az ember a változás Kübler-Ross görbéjén, és mindent meg kell tennie, hogy beosztottjai, munkavállalói minél könnyebben tegyék meg ugyan ezt az utat.

- A **kultúra, sport, munkahelyi jóllét** tárgyak esetében a munkahelyi jól-lét az, amit érint a szakdolgozatom. 2024-ben is rengeteg olyan folyamattal lehet találkozni egy nagyvállalat esetében, ahol szakemberek, sok éves tapasztalattal és tudással adatrögzítő feladatokat látnak el, és adatokat másolnak egyik rendszerből a másikba, hiányzó szolgáltatások vagy komponensek miatt. Ezen a ponton jogosan kérdezik meg, hogy valóban ez az a munka, amit végezniük kell, nincs-e olyan értékteremtő, vállalati előnyt biztosító tevékenység, amit végezhetnének. Ezt a folyamatot languishing-nek, magyarul legjobban talán senyvedésnek, vagy sorvadásnak nevezhetjük, amikor ezek az emberek nem tudják kihasználni lehetőségeiket, képességeiket, és lassan csak elfogadják a munkakörükkel járó monoton, kevés üzleti értéket képviselő feladatok elvégzését. Véleményem szerint ez a folyamatot legalább olyan fontos megállítani, mint a kiégést, és a kreatív, értéket teremteni akaró munkavállalókat felszabadítani a monoton, lélekölő és agyzsibbasztó folyamatok elvégzése alól.
- A **szakterületi jogi ismeretek** tárgy több szinten is megjelenik a dolgozatban. Egyszer, mint az RPA szoftver liszenszelésének részeként, miszerint megfelelő liszenszkulcsot használ-e egy vállalat az automatizációjához, például használ-e fejlesztői liszenszt éles üzemeltetésre. Másrészt a robotikus folyamatok fejlesztése, illetve tesztelése alatt valós ügyfeladatok felhasználása tekintetében használ-e a vállalat megfelelő anonimizálást az adatokat, van-e megfelelő teszt rendszer, egyáltalán, hogyan menedzseli a cég az ügyfeladatokat (GPDR). Harmadszor pedig van-e olyan folyamat, ami jogi szempontból nem automatizálható, akár olyan esetben, hogy szerződéses partner tiltja, hogy az általuk üzemeltetett rendszerben robotok dolgozzanak.
- Az **elektronika fizikai alapjai** és az **elektronikus áramkörök** tantárgyat érintem a robotokhoz szükséges fizikai hardware infrastruktúra kialakítása részeként.
- A **matematikai alapok** a szakdolgozat működő robotikus folyamatában játszik szerepet, hiszen a kódolás alapvetően a matematikai logikára, gondolkodásra épít. Elég csak Boole algebrára gondolni. Ahhoz, hogy egy üzleti logikát helyesen tudjuk leképezni egy szoftver robottal, tisztában kell lennünk a logikai műveletekkel, a ciklusokkal, de egy folyamat lépéseit akár az „utazó ügynök problémájaként” is felfoghatjuk és a robotunk máris kombinatorikai térben

(RPA_Excel_Kombinatorikai_tér.xlsx csatolmány) találja magát, hogy mi a legoptimálisabb feladatvégzés (ami sok esetben nem ugyan az a lépés sorozat, amit emberileg véghez viszünk, akár a kényelem vagy a megszokás miatt).

- Az **adatszerkezetek és algoritmusok** gyakorlatban bármilyen robotikus folyamatban, bármilyen lépésben jelen vannak. Képzeljük el, hogy egy terminál képernyőjéről olvas be a robot adatokat, amik több oldalon keresztül vannak felsorolva, majd kiolvasás után ezeket az adatokat sorba kell rendeznünk, majd ezeket feldolgozni tömbként vagy listaként.
- A **hálózatok és számítógép architektúrák** fontos részét képezik az implementálandó RPA hálózatnak. Hány gépet tervezünk használni, ezek távoli virtuális gépek, esetleg fizikai eszközök egy lezárt szerverszobában. Valamint érintőlegesen említem az Orchestrator telepítése kapcsán.
- Az **operációs rendszerek** tárgy még részletesebben megjelenik, nem csak az Orchestrator szerveren futó operációs rendszer miatt, hanem a virtuális gépen futó OS miatt. Részletezem miért fontos az összes robot gépen, hogy ugyan az az OS legyen telepítve, ugyan azokkal a paraméterekkel és beállításokkal, mert beláthatatlan hibákat tudnak okozni apró különbségek is.
- A **programozás és programozási alapelvek és módszertanok** teljes egészében részét képezik a robot létrehozásában, akár, ha egy sorbarendezezésről beszélünk, akár bonyolultabb kódrészletekről beszélünk. Az RPA ugyanakkor nagyban épít a kész, külső komponensek alkalmazására, ebben nagy segítséget ad maga a UiPath is mint vállalat, akik rengetek olyan csomagot készítenek, amik a műveleteket könnyebbé teszik az átlagos felhasználónak, aki így kódolás nélkül, drag-and-drop módszerrel tudnak például Salesforce Lightning API-t használni, de ez csak egy példa a sok közül.
- A **rendszermodellezésként** a követelményspecifikáció létrehozása, ahol az AS-IS folyamat dokumentálásra kerül, ahol az üzleti oldal által végzett folyamatlépéseket alacsony szinten dokumentáljuk, kvázi kattintás szinten, ahol minden egyes döntési pontot adatokkal kell alátámasztanunk. Például egy keresés esetén mi lehet az eredmény? Mi történik, ha nincs találat, mi történik, ha egy találat van, szükséges-e azt validálni, vagy mi történik akkor, ha több találatot kapunk? Szükséges-e választani, vagy ez már önmagában hibát jelent?
- A szoftverarchitektúrák tárgy és a **rendszertervezés** pedig a megvalósítási tanulmány (angolul Process Definition Document, PDD) eredménye, ahol megtervezzük a robot

alapvető működését. Hány egységből fog állni, tabuláris adatokkal dolgozik vagy queue itemekkel. Szükség van-e az üzlet számára az adatok riportálására, és ha igen milyen formában.

- Az **adatbázisok** megjelennek akár a robot által létrehozott logok vagy queue item-ek tárolásában, de magának a robot működésének a részét is képezhetik, ha adatbázisokból kell kinyernie a robotnak a feldolgozandó tételek listáját. Ebben az esetben a pénzügyi világ alapját jelenti Excel is adatbázisnak tekinthető, ugyanis a robot számára a strukturált adat, adattáblák jelenik a fő bemeneti információt.
- A **szoftvertesztelés** része a fejlesztési folyamatban, akár fejlesztői tesztről, akár üzleti elfogadási tesztről (UAT, user acceptance test), élesítés utáni visszatesztelésről vagy regressziós tesztről beszélünk.
- A **szoftverüzemeltetés** szerves része a CoE-nek (Center of Excellence, kompetencia központ). Mivel a legtöbb folyamat emberi inputra épül, sok esetben történhet hiba az adatok előállításában, ami szükségessé teszi a robotok jelzésének monitorozását, az üzleti területek értesítését, a folyamatos kommunikációt és a folyamatok újraindítást. A meglévő erőforrásokat is figyelembe véve optimális állapotot kell találni a folyamatok ütemezésében is.
- A **felhasználói interfészek és vizualizáció** leginkább a robotok által készített riportok tekintetében jelenik meg. A leggyakrabban kért adatok a robot által feldolgozott tételek száma hisztogram formában, és ezek bármilyen további bontása rendelkezésre álló paraméter alapján. De ugyan ilyen lehet a ciklusidő vagy a hibás tételek megoszlása a teljes volument tekintve, vagy az üzleti kivételek számának monitorozása.
- Az **informatikai védelem és biztonság** a robotok jogosultságának menedzselésében jelenik meg, hogy a különböző jogok és rendszerhozzáférések igénylése és számontartása miként történik meg, auditálható formában.
- A **Tudásmenedzsment az IT-biztonság területén** tárgy a robotok dokumentációval kapcsolatos részén van kiemelve, ahol a megvalósítása tanulmány szolgál, mint kollektív tudás, ahol a robotokhoz tartozó tudás/kód ezáltal nincs limitálva csak az adott fejlesztőre (*The S.M.E. is dead, Long live collaborative intelligence¹*)
- Az **innovatív információs és kommunikációs technológiák az IT-biztonság kapcsán** tárgy gyakorlatilag esernyője alá tartozik a teljes szakdolgozat, mivelhogy maga az RPA

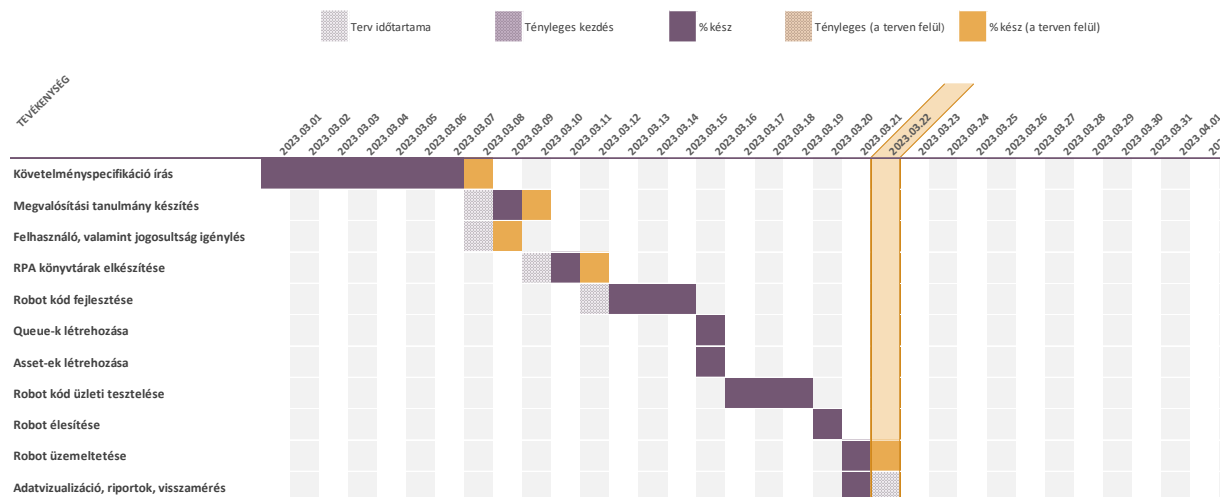
1 <https://ritchiediopita.wordpress.com/2018/03/26/journal-1-subject-matter-expert/>

is innovatív technológiának tekinthető, és ezen technológia ismertetése a szakdolgozatom témája.

- Az **IT-biztonsági fejlesztések minőség- és projektmenedzsmentje** tárgy a robotok részletes, minden kimenetelt tartalmazó visszatesztelése alkalmával van érintve.
- A **mesterséges intelligenciák az IT-biztonság területén** tárgyat pedig azáltal érintem miszerint nevezhetünk mesterséges intelligenciának egy olyan autonóm működést, ahol az input átadásán túl nincs emberi érintkezés a robot működése közben. Mi történik, ha az üzleti tudás, akár csak lustaságból, vagy emberi mulasztásból kihal a robot mögött, és az emberi elvándorlás miatt gyakorlatilag az üzleti terület már nem ismeri a saját folyamatát, mert csak annyi a dolga, hogy engedje a robotot dolgozni. Ebben az esetben valós intelligenciának nevezhetjük-e az emberi intelligenciát? Valamint szakdolgozatom kivonatát is a ChatPDF (<https://www.chatpdf.com/>) segítségével készítettem el, angol nyelvű fordítását pedig a ChatGPT-vel.

3. Saját fejlesztések

A saját fejlesztések fejezet a vállalati környezetben használt szoftverfejlesztés életciklusa alapján lett meghatározva, kronologikus sorrendben. A fejlesztés életciklusa az alábbi Gantt chart-on látható.



3. ábra Gantt chart (Forrás: Gantt-RPA.xlsx melléklet - saját közlés)

A fejezetekhez tartozó forráskódok és dokumentumok általam lettek létrehozva.

A ChatGPT határait már ott elértem, hogy képtelen volt olyan XAML fájlt előállítására, amit a Studio képes lett volna megnyitni.



You

Készíts egy olyan UiPath által megnyitható xaml kódot, ami egy 10 ismétlődésű for ciklusban kiírja az adott ismétlődés számát



ChatGPT

Az alábbi példa egy UiPath Workflow fájlt mutat, amely egy 10-szeres ismétlődésű for ciklust tartalmaz, és minden egyes iterációban kiírja az adott iteráció számát a kimeneti ablakba (`Output` panel). A fájl kiterjesztése `.xaml`.

```
xml
Copy code

<?xml version="1.0" encoding="utf-8"?>
<Activity mc:Ignorable="sap sap2010 sads" x:Class="Main" mva:VisualBasic.Settings="As
  <TextExpression.NamespacesForImplementation>
    <sco:Collection x:TypeArguments="x:String">
      <x:String>System.Activities</x:String>
      <x:String>System.Activities.Statements</x:String>
      <x:String>System.Activities.Expressions</x:String>
      <x:String>System.Activities.Validation</x:String>
      <x:String>System.Collections.Generic</x:String>
      <x:String>System.Data</x:String>
      <x:String>System.Diagnostics</x:String>
      <x:String>System.Drawing</x:String>
      <x:String>System.IO</x:String>
      <x:String>System.Linq</x:String>
      <x:String>System.Net.Mail</x:String>
      <x:String>System.Net.Mime</x:String>
      <x:String>System.Reflection</x:String>
      <x:String>System.Text</x:String>
      <x:String>System.Text.RegularExpressions</x:String>
      <x:String>System.Threading.Tasks</x:String>
      <x:String>System.Xml</x:String>
      <x:String>System.Xml.Linq</x:String>
    </sco:Collection>
```

4. ábra ChatGPT által készített xaml állomány (Forrás: <https://chat.openai.com/c/88e3ba2e-6af2-41f1-b049-91f41b14e9f5>, Folyamatszervezés riport keretrendszer\Performer_DB.Activities\ ChatGPT_xml.xaml melléklet)

Ez a kód az alábbi hibát adja megnyitáskor: 'sap2010' prefix is not defined. Line 1, position 11. Valamint a későbbiekben bemutatott titkosításra használt VBA makró elkészítésével is akadtak gondja.



You

Készíts egy olyan VBA makrót, ami bemeneti értékek alapján képes az alábbi Excel beállítások elvégzésére egy kész fájl.

Füzetvédelem beállítása

Megnyitás-olvasásra (jelszó használata)

Megnyitás csak jelszóval (jelszó használata)

Bemeneti értékek: jelszó mint string, fuzevedelem mint boolean, megnyitas_olvasasra mint boolean, megnyitas_csak_jelszoval mint boolean.



ChatGPT

Az alábbi VBA makró elvégzi az általad megadott beállításokat egy kész fájl. A bemeneti értékek a következők:

- ``password`` (jelszó mint string)
- ``protectSheet`` (füzetvédelem mint boolean)
- ``openReadOnly`` (megnyitás-olvasásra mint boolean)
- ``openWithPasswordOnly`` (megnyitás csak jelszóval mint boolean)

```
vba Copy code

Sub SetExcelSettings()

    Dim filePath As String
    Dim password As String
    Dim protectSheet As Boolean
    Dim openReadOnly As Boolean
    Dim openWithPasswordOnly As Boolean

    ' Bemeneti értékek beállítása
    filePath = "C:\Path\to\Your\File.xlsx" ' Módosítsd a fájl elérési útvonalára
    password = "YourPassword" ' Add meg a jelszót
    protectSheet = True ' Igazra állítva bekapcsolja a füzetvédelmet
    openReadOnly = True ' Igazra állítva csak olvasásra nyitja meg a fájlt
    openWithPasswordOnly = True ' Igazra állítva csak jelszóval nyitja meg a fájlt
```

5. ábra ChatGPT által készített VBA makró (Forrás: <https://chat.openai.com/c/51b3f616-cc2b-49fa-9814-259d852059af>, ChatGPT_VBA.txt melléklet)

A megnyitás olvasásra beállítást úgy érzékelt, mintha a fájl maga lenne read only, nem pedig lementésnél állította be, hogy csak jelszóval lehessen megnyitni.

3.1. Követelményspecifikáció

Ennek a státusznak (követelményspecifikáció írás) a hossza lehet a többi résznél is hosszabb, akár mindennél együtt is több időt igényel a dokumentum elkészítése. A robotnak az alábbi követelményeknek kell eleget tenniük. Bármilyen, ami a követelményspecifikációban nem szerepel,

az nem része a robot működésének. Egy rosszul, nem elég részletesen meghatározott követelményspecifikáció a változtatási igények végtelennek tűnő ismétlődését képes elindítani.

- A robot képes legyen az adattárház meghatározott sémáiból és tábláiból kiolvasni a feldolgozandó kimutatásokat (**input tábla**)
- A robot küldjön hibaüzenetet, ha az adott séma, adott táblája üres.
 - Üzenet címzettje: „johndoe@myltd.com”
 - Üzenet tárgya: **séma** + „.” + **tábla** + „,üres”
 - Üzenet feladójának neve: „Folyamatszervező Robot”
 - Üzenet feladójának e-mail címe: folyszerv.kimutatas.robot@myltd.com
 - Üzenet szövege: „A tárgyban megadott tábla üres. Ellenőrizd ennek helyességét, és adott esetben a robot üzemeltetéstől kérj új, eseti futtatást.”
- A robot minden esetben írjon vissza az input táblába, hogy az adott sor kimutatása sikeresen vagy hibásan lett feldolgozva, mi az eredmény, valamint töltse, mikor kezdett el dolgozni az adott sorral és mikor fejezte be.
- A robot képes meghatározott adattáblát kimenteni XLSX formátumban az adattárházból
- XLSX esetében a robot legyen képes maximum 10 darab kimutatást egy fájlba tölteni
- A robot képes meghatározott adattáblát kimenteni CSV formátumban az adattárházból
- CSV esetén paraméterezhetően meg lehessen adni, hogy vessző, pontosvessző vagy tab legyen a szeparátor
- A fájl neve, valamint a fül(ek) neve szintén paraméterizálható legyen
- XLSX esetében a robot képes legyen lapfülvédelem beállítására, szintén paraméterezhetően
- XLSX esetében a robot képes legyen jelszóval ellátni a fájlt megnyitásra és/vagy szerkesztésre. Ebben az esetben a „Folyszerv.Terület” jelenjen meg a felugróablakban amikor az írási jelszót kéri az Excel
- XLSX esetében az alábbi formázás(oka)t legyen képes elvégezni a robot:
 - Automatikus oszlopszélesség beállítása
 - Adott oszlop rögzítése
 - Adott sor rögzítése
 - Autofilter elhelyezése
 - Nagyítás beállítása
 - Fájl megnyitásakor melyik fül jelenjen meg alapértelmezetten
 - Kimutatás adattartalom szövegformátumba alakítása

- A robot képes legyen a kimutatás(oka)t megadott hálózati meghajtóra, vagy emailben elküldeni
- Email esetén az alábbiak paramétereizhetők legyenek:
 - Feladó neve
 - Feladó e-mail címe
 - Címzett címe
 - Másolatot kapó címzett
 - Titkos másolatot kapó címzett
 - Tárgy
 - Levéltörzs (alap html formázást támogatva)
- Email küldése esetén a robot megadott fájl méret felett ne küldje el a kimutatást, hanem jelölve meg a feldolgozandó kimutatást üzleti kivételnek.

Név	Leírás	Vezérlő elem, input (ha töltve van)	Kötelező	Értékszlet	CSV esetén nem értelmezett
id	Egyedi azonosító (rekordazonosító)		x		
PAID	Csomagazonosító		x		
GRID	Rekord azonosító		x		
DWH_TABLE_01	Exportálandó tábla 1	x	x		
SHEET_NAME_01	Lapfűl 1 neve	x	x		
SELECT_01	Select 1 megadása	x	x		
DWH_TABLE_02	Exportálandó tábla 2	x			x
SHEET_NAME_02	Lapfűl 2 neve	x			x
SELECT_02	Select 2 megadása	x			x
DWH_TABLE_03	Exportálandó tábla 3	x			x
SHEET_NAME_03	Lapfűl 3 neve	x			x
SELECT_03	Select 3 megadása	x			x
DWH_TABLE_04	Exportálandó tábla 4	x			x
SHEET_NAME_04	Lapfűl 4 neve	x			x
SELECT_04	Select 4 megadása	x			x
DWH_TABLE_05	Exportálandó tábla 5	x			x
SHEET_NAME_05	Lapfűl 5 neve	x			x
SELECT_05	Select 5 megadása	x			x
DWH_TABLE_06	Exportálandó tábla 6	x			x
SHEET_NAME_06	Lapfűl 6 neve	x			x
SELECT_06	Select 6 megadása	x			x
DWH_TABLE_07	Exportálandó tábla 7	x			x
SHEET_NAME_07	Lapfűl 7 neve	x			x
SELECT_07	Select 7 megadása	x			x
DWH_TABLE_08	Exportálandó tábla 8	x			x
SHEET_NAME_08	Lapfűl 8 neve	x			x
SELECT_08	Select 8 megadása	x			x
DWH_TABLE_09	Exportálandó tábla 9	x			x
SHEET_NAME_09	Lapfűl 9 neve	x			x
SELECT_09	Select 9 megadása	x			x
DWH_TABLE_10	Exportálandó tábla 10	x			x
SHEET_NAME_10	Lapfűl 10 neve	x			x
SELECT_10	Select 10 megadása	x			x
FILE_FORMAT	Fájl formátum	x	x	XLSX, CSV	
FILE_CSV_SEPARATOR	CSV esetén szeparátor	x		VESSZŐ, PONTOSVESSZŐ, TABULÁTOR	
FILE_NAME	Fájl nevének megadása	x	x		
ENCRYPTION	Kell fűzetvédelem - titkosítás?	x	x	IGEN, NEM	
PASSWORD	Jelszó	x			
READING_WITHOUTPASSWORD	Olvasásra megnyitható jelszó nélkül?	x		IGEN, NEM	
READING_ONLYWITHPASSWORD	Olvasásra megnyitható jelszóval?	x		IGEN, NEM	
DATATYPE_TEXT	Minden mezőnél adattípus átállítása szövegre	x		IGEN, NEM	x
AUT_COLUMN	Automatikus oszlopszélesség	x		IGEN, NEM	x
FIX_ROW	Rögzített sor száma	x			x
FIX_COLUMN	Rögzített oszlop száma	x			x
AUT_FILTER	Automatikus szűrő	x		IGEN, NEM	x
LARGE	Nagyítás mértéke	x			x
FIRST_PAGE	Alap lapfűl megnyitáskor	x			x
TYPE	Hálózati vagy email küldés	x	x	EMAIL, NETWORK	
MAPPA	Hálózati mappa megadása	x			
SENDER	Email küldő megadása	x			
ADDRESS	Email címzettek megadása	x			
CC	Emailben CC megadása	x			
BCC	Emailben BCC megadása	x			
SUBJECT	Email tárgy megadása	x			
TEXT	Email törzs megadása (max. 2000 karakter)	x			
LIMIT	Fájl méretének korlátozása	x	x		
RPA_START	RPA tölti -> RPA feldolgozás kezdő időpontja		x		
RPA_END	RPA tölti -> RPA feldolgozás záró időpontja		x		
RPA_STATUS	RPA tölti -> RPA feldolgozás - Státusz kód (Hiba vagy sikeres feldolgozás)		x		
RPA_ERCODE	RPA tölti -> RPA feldolgozás - Státusz leírás / Hiba leírás		x		

6. ábra Input adattábla szerkezete és mezők magyarázatai (Forrás: RPA_Excel_Kombinatorikai_tér.xlsx melléklet, RAW lapfűl – saját közlés)

3.2 Megvalósítási tanulmány

Ebben a fejezetben a robot kód magas szintű bemutatása szerepel, a készítendő al-robotok száma és fő funkciója, és keretrendszer meghatározása.

2 robot kód:

- **Dispatcher** keretrendszer a tabuláris adatokhoz: Input tábla (WRK_RPA_EXCEL_EMAIL) olvasása megadott sémából, és queue-ba töltés. Input tábla frissítése egységesen, ha az adatokat sikeresen elhozta a robot.
- **Performer** keretrendszer a queue item-ekhez: a queue-ból a feldolgozandó tételek(riportok) elkészítése, hálózati meghajtóra mentése/SMTP-n való kézbesítése,

Excel fájlok esetén formai formázás és/vagy titkosítás. Tételenként való input táblába írás.

- Előkövetelmény: REFramework használata: a keretrendszer adottság, ezt felhasználjuk, a keretrendszer xaml fájljait töltjük fel adatokkal.

Queue

- Folyamatszervező_riportok

Asset

- FolySzerv_EmailTo: string, címzettek email jelzéshez üres tábla esetén
- DWH_Credential: adattárház hozzáférés felhasználóneve és jelszava

General Problem Solving

A létrehozandó dispatcher robot automatizációs potenciálja egy olyan új sablon, ami bármilyen, dinamikusan változtatható adattáblából képes beolvasni bármilyen adatot, és azt queue itemmé alakítja. Ezzel bármilyen jövőbeli folyamat automatizációját jelentősen felgyorsíthatjuk, sőt az egységet robot input fogalmát is megteremtettük. Amennyiben a DWH connector helyett saját megoldást használunk (például Access adattáblát), akkor a connector helyett bármilyen saját kódot is felhasználhatunk az adatok beolvasásához.

3.2.1. Dispatcher

Az alapvető működés és state-ek közötti mozgás.

3.2.1.1. Init

- "Framework\KillAllProcesses.xaml"
 - excel.exe folyamat terminálása
- "Framework\InitAllSettings.xaml"
 - Config dictionary létrehozása a "Data\Config.xlsx" alapján, "Settings" és "Constants" fűlekről
 - out: Config dictionary
- "Framework\InitAllApplications.xaml"
 - Adattáblához való kapcsolódás
 - Input táblából való adatleválogatás lokális fájlba
 - Adattáblától való lekapcsolódás
 - in: Config dictionary

- in: DWH scheme: kód szintű változó, paraméterezzhető, hogy milyen sémában kell futnia a robotnak
- Átmenetek
 - Init hiba esetén, ha a max init számláló nem érte el a maximumot → Init
 - Init hiba esetén, ha a max init számláló elérte a maximumot → End
 - Sikeres iniciálás esetén → Get Transaction Data

3.2.1.2. Get Transaction Data

- Stop szignál ellenőrzése
 - Ha Orchestratorból a folyamat le lett állítva, a folyamat ennél a lépésnél megáll, azaz a következő tétel = Nothing
- "Framework\GetTransactionData.xaml"
 - Ha nem létezik még az adattábla változó, akkor "Process\ReadTransactionData.xaml" meghívása akkor:
 - Lementett lokális fájl beolvasása adattábla változóba
 - Ha az adattábla változó sorainak száma nulla, jelzés küldése az üzlet számára
 - Ha már létezik, akkor új sor lekérése az adattábla változóból, mint feldolgozandó tétel (egy datarow változó). Ellenkező esetben a tétel = Nothing
 - Hiba esetén következő tétel = Nothing
 - Átmenetek
 - Ha feldolgozandó tétel = Nothing → End
 - Sikeres tétellehívás esetén → Process Transaction

3.2.1.3. Process Transaction

- "Process.xaml"
 - Az aktuális datarow-t queue item-ként feltöltjük az Orchestrator queue-ba (így több robot is tud majd egyszerre dolgozni a kimutatások elkészítésében). A datarow kiegészítjük a korábban használt séma adatával, így a performer egység tudja melyik sémába kell visszaírni az adatokat.
 - in: TransactionItem (mint datarow)
 - in: Config dictionary (ez tartalmazza a queue nevét)
 - in: DWH séma
- "Framework\SetTransactionStatus.xaml"

- Ha a Process.xaml-ben váratlan hiba történik, akkor képernyőkép készítése a hiba pillanatáról (config dictionary-ben megadott elérési útvonalra)
- Ha van újrapróbálkozási működés, akkor a tételt ismétlése
- Ha van egymás utáni maximalizált hibák száma működés, akkor ennek figyelése
- Átmenetek
 - Sikeres feldolgozás esetén → Get Transaction Data
 - Üzleti kivétel esetén (várt hiba) → Get Transaction Data
 - Hibás feldolgozás esetén (nem várt hiba) → Init
 - Ha elérte a robot a maximális, egymás utáni hibák számát → End
 - Ha hiba történt a "Framework\SetTransactionStatus.xaml"-ben → End

3.2.1.4. End

- Ha hiba történt, és ez a hiba nem a "Framework\Process.xaml"-ben történt, akkor hiba küldése az RPA üzemeltetés felé
- "Framework\CloseAllApplications.xaml" meghívása
 - Ha ez hibára megy, akkor "Framework\KillAllProcesses.xaml"

3.2.2. Performer

Az alapvető state-ek közti mozgás megegyezik a dispatcher működésével. A különbség a GetTransactionData state-ben van, innen nem datarow-t adunk tovább, hanem queue item változót, mivel itt már nincs Excel adattábla input, csak az Orchestrator queue-ba feltöltött queue item(ek).

3.2.2.1. Init

- "Framework\KillAllProcesses.xaml":
 - excel.exe
- "Framework\InitAllSettings.xaml"
 - out: Config dictionary
- InitAllApplication:
 - DWH DB kapcsolat létrehozása
 - in: Config
 - out: DWH_DB

3.2.2.2. GetTransactionData

- Standard framework, új queue item lehívása
 - out: TransactionItem

3.2.2.3. Process Transaction

- TempReports (C:\Temp\TempReports mappa) törlése és újra létrehozása (korábban lementett fájlok törlése)
- For ciklus, 10 ismétlés (Enumerable.Range(0,10).ToArray(), mivel az input táblában up-to 10 riport készülhet el egy Excel esetén)
 - SQL select parancs meghatározása:
in_TransactionItem.SpecificContent("SELECT_"+(n+1).ToString("00")).ToString
--ahol n a for ciklus ismétlődési indexe
 - Sheet név meghatározása:
in_TransactionItem.SpecificContent("SHEET_NAME_"+(n+1).ToString("00")).ToString
 - Ha az adott SQL select parancs érték üres, akkor kilép a robot a ciklusból, mert az adott sorban nincs (több) lehívandó riport
 - Ha az SQL select parancs nem üres, akkor a riport lehívása megtörténik:
 - TableDownload.xaml - Invoke Workflow File
 - SQL select query futtatása
 - Ha a FILE_FORMAT = xlsx, akkor Excelben történő mentés kerül meghívásra:
 - FilePath: TempReports + fileName + fileExtension
 - SheetName: Sheet név
 - Cell: A1
 - AddHeaders: True
 - Ha a FILE_FORMAT nem xlsx (feltételezés, hogy csak csv lesz, ha nem xlsx):

- SEPARATOR alapján csv lementése; elfogadott szeparátorok: "vessző", "pontosvessző", "tab"
 - FilePath: TempReports + fileName + fileExtension
 - AddHeaders: True
- A 10-es ciklus esetében működésbeli feltétel, hogy csak xlsx esetén lehet több mint 1 SQL select; csv esetén ez technikailag nem tud működni
 - xlsx esetén pedig mindig ugyan az a fájl lesz felülírva, új lapfül beszúrásával.
 - Feltételezés, hogy a sheet name érték minden esetben egyedi lesz
- Riportok lehívása után és xlsx esetén VBA makrók futtatása a formázás és titkosítás elvégzésére:
 - formatting: Sub formatting(text_data, autowidth, fixrow, fixcol, ApplyFilter, ZoomLevel, lapfülszámaforopen)
 - titkosítás: Sub titkositas(incr, pw, withoutpw, onlywithpw)
- TRANSFER_TYPE megállapítása
 - EMAIL
 - fileMéret meghatározása (lementett kimutatás mérete MB-ban)
 - ha nincs maximalizált limit határérték (FILE_SIZE_LIMIT), akkor a fileMéretLimit= fileméret+1
 - Ha a fileMéret > fileMéretLimit
 - New BusinessException("A megadott fájl már nem küldhető ki a mérete miatt")
 - Email küldése SMTP-n paraméterek alapján
 - HTML Body: True
 - Ha az e-mail küldés hibára megy, akkor az ideiglenes fájl törlése + New BusinessException(exception.Message)
 - NEM EMAIL (tehát mappába mentés, azaz NETWORK érték)

- a TempReports mappából a valós helyre mozgatása a riportnak
(in_TransactionItem.SpecificContent("FOLDER").ToString+ "\" +
fileName + "." + fileFormat)
- ha a FOLDER utolsó karaktere "\" akkor azt a robot levágja, és maga
füzi hozzá
- Feltételezés, hogy a robotnak nem lesz szüksége felülírni hálózati
meghajtóra mentés esetén semmilyen fájlt (overwrite property: false)
- Riportonként INPUT SQL tábla frissítése:
 - "UPDATE
"+TransactionItem.SpecificContent("DWH_Source").ToString+".WRK_RPA_EX
CEL_EMAIL SET RPA_STATUS = '"+Status+"', RPA_ERCODE = '"+Reason+"',
RPA_START = '"+TransactionStartTime.ToString("yyyyMMddHHmmss")+"',
RPA_END =TO_CHAR(CURRENT_TIMESTAMP, 'YYYYMMDDHH24MIss')
WHERE ID = '"+TransactionItem.SpecificContent("ID").ToString+"'"
 - Ez soronkénti commitot is jelent

3.2.2.4. End

- CloseAllApplications:
 - lecsatlakozás a DWH DB-ről
 - in: DWH_DB

3.3 Felhasználó, valamint jogosultság igénylés

Ahhoz, hogy a robot be tudjon lépni a szükséges rendszerekbe (adattárház), szükséges egy egyedi felhasználói azonosítóval rendelkeznie, amire a jogosultságot meg lehet igényelni. Ezt a feladatot a helpdesk látja el, a feladat a szükséges kód létrehozása, és a korábban létrehozott robot munkaköri sablon alkalmazása (ez a sablon van arra használva, hogy a korábban igényelt mappáhozáférések minden robot esetén elérhetőek legyen, bármilyen gépen is futtatunk bármilyen folyamatot.

Az azonosító megléte után, megigényeltem a DWH jogosultságot is.

3.4. RPA könyvtárak elkészítése

Az igényhez előzetesen 2 olyan szükséges könyvtárat is megállapítottam, amit a kódban többször is felhasználunk, valamint más robotok esetében is felhasználhatunk a későbbiekben.

Az egyik a DWH-val kapcsolatos adatkiépítés (ahol a bementi paraméter az Orchestrator-ban tárolt asset neve, kimeneti értéke pedig maga a database data connection), a másik pedig az email küldés, SMTP-n keresztül.

3.4.1. SMTP Connector Könyvtár

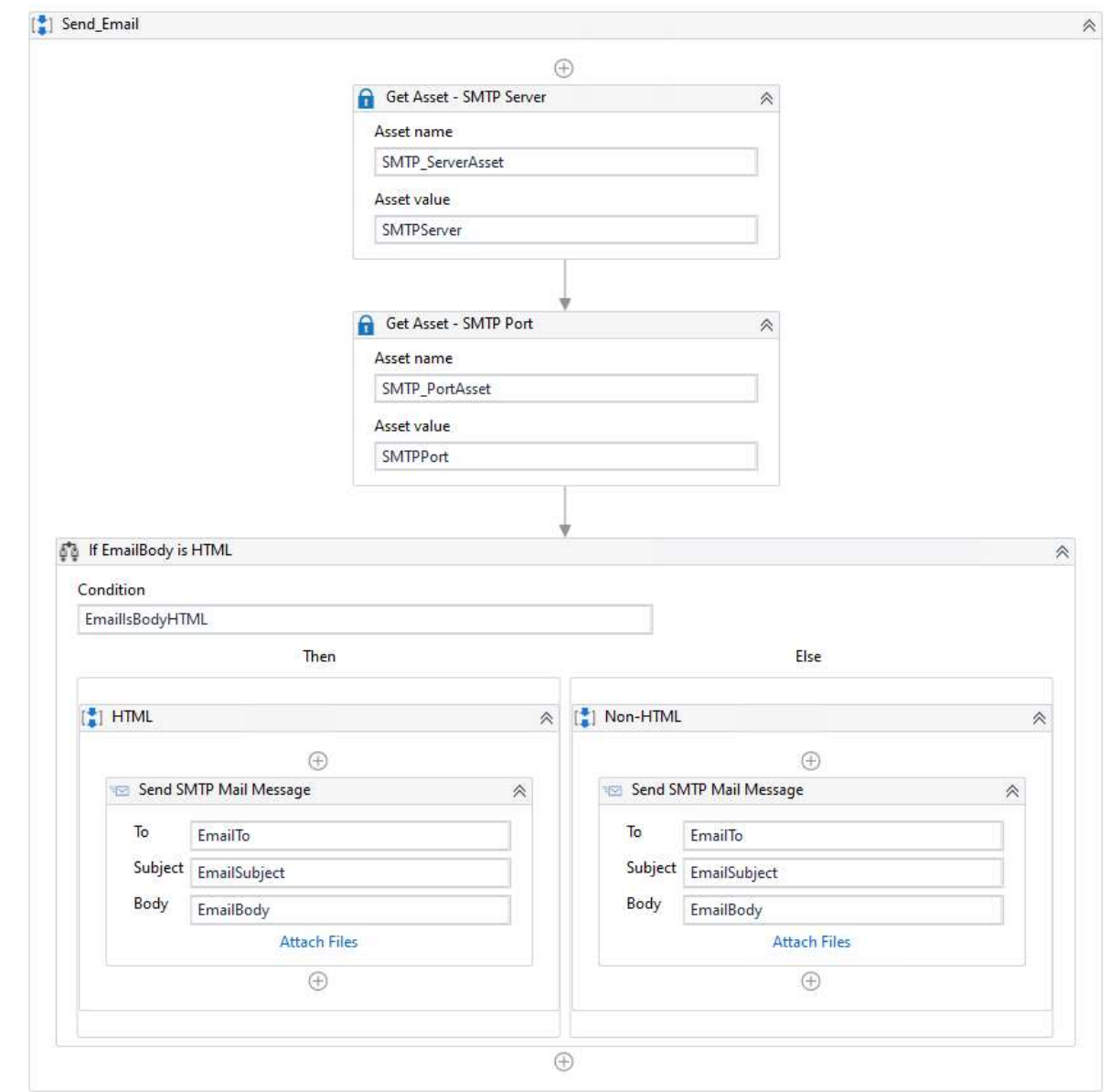
Library_Emailing és SMTP_Connector.1.0.1.nupkg csatolányok.

Összesen 4 eseményt határoztam meg, amikor e-mailt küldhetünk.

- Sima email
- Email egy csatolmánnyal
- Email több csatolmánnyal
- Email képernyőképpel

Sima email esetében szükségünk lesz a céges SMTP server és port adatokra, amiket az Orchestrator-ban tárolunk. Szükség esetén, ha ezek az adatok változnak, központilag egy helyen elég frissíteni őket, és az összes függő robot azonnal az új adatokat használja majd.

Az EmailsBodyHTML változó határozza meg, hogy az e-mail törzse HTML formátumban van írva, vagy sem. Más különbség nincs a két activity között.



7. ábra SMTP email küldés csatolmány nélkül (Forrás: Library_Emailing\SMTP\ Send_Email.xaml melléklet – saját közlés)

Name	Direction	Argument type
SMTP_ServerAsset	In	String
SMTP_PortAsset	In	String
EmailFromName	In	String
EmailFromAddress	In	String
EmailTo	In	String
EmailCC	In	String
EmailBCC	In	String
EmailBody	In	String
EmailIsBodyHTML	In	Boolean
EmailSubject	In	String

8. ábra Bemeneti értékek és típusaik (Forrás: Library_Emailing\SMTP\ Send_Email.xaml melléklet – saját közlés)

Ha nevesített csatolmánnyal szeretnénk küldeni ugyan ezt az emailt, akkor a kettő közt csupán annyi a különbség, hogy utóbbi bemeneti paraméterként a csatolandó fájl teljes elérési útvonalát is várja, mint string, majd azt a Send SMTP Mail Message activity attachments property-jébe is elhelyezi.

Több csatolmány esetén pedig nem string-et nem, hanem IEnumerable<string> értéket, amit előtte kell feltölteni a csatolmányok teljes elérési útvonalával.

Képernyőkép csatolása esetén pedig nem szükséges külön meghatározni a képernyőkép elérési útvonalát, azt xaml maga elkészíti, ideiglenesen lementi, csatolja, mint egy csatolmány esetén, majd küldés után az ideiglenes fájlt törli.

3.4.2. DWH Connector Könyvtár

Library_DWH és DWH_Connector.1.0.1.nupkg csatolányok.

Adattárház kapcsán pedig két könyvtár elemet hoztam létre, a ChatGPT-vel is egyeztetve.



You

hogyan kapcsolódhat egy UiPath Robot egy oracle adatbázishoz?



ChatGPT

UiPath Robotnak lehetősége van kapcsolódni egy Oracle adatbázishoz többféle módon, például az alábbiak szerint:

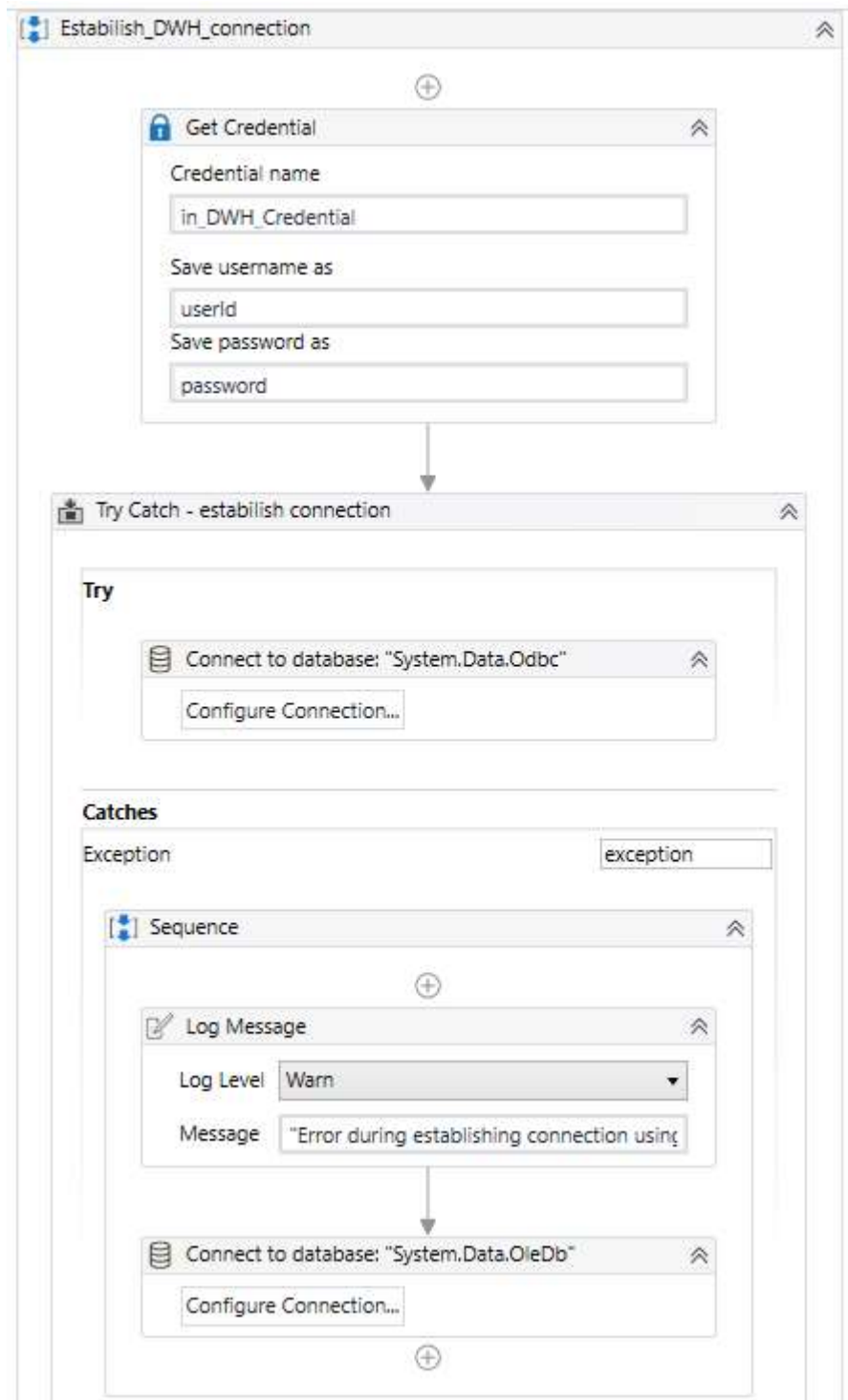
1. **UiPath Database Activities Package használata:** Az UiPath Studio-ban elérhető Database Activities Package segítségével könnyedén kapcsolódhat egy Oracle adatbázishoz. Ez a csomag olyan tevékenységeket tartalmaz, mint a Connect, Query, Insert, Update, Delete, stb. Ezeket a tevékenységeket használva a Robot képes SQL-lekérdezéseket futtatni az adatbázisban, adatokat beszúrni, frissíteni vagy törölni.
2. **ODBC vagy JDBC Driver használata:** Ha az Oracle adatbázishoz kapcsolódásához ODBC vagy JDBC Driver szükséges, akkor telepítheti és konfigurálhatja ezeket a drivereket a környezetben, majd az UiPath Studio-ban az „Invoke Code” tevékenységet használva írhat egy kis kódot (C# vagy VB.NET), amely kapcsolatot hoz létre az adatbázissal és végrehajtja a szükséges műveleteket.
3. **UiPath Orchestrator használata:** Ha az Oracle adatbázis elérhető egy másik rendszerből, például egy szerveren vagy felhőalapú platformon, akkor a UiPath Robot képes lehet az Orchestrator segítségével elérni ezt az adatbázist. Ehhez konfigurálni kell az Orchestrator környezetet és a Robot gépét a megfelelő hozzáférési engedélyekkel.
4. **Egyéb kiegészítő modulok vagy API-k:** Ha rendelkezésre állnak olyan speciális kiegészítő modulok vagy API-k, amelyek lehetővé teszik az Oracle adatbázissal való kommunikációt, akkor ezeket az UiPath alkalmazásokban is fel lehet használni. Ehhez azonban gyakran szükség van fejlesztési ismeretekre vagy külső szakértők bevonására.

Fontos megjegyezni, hogy az adatbázishoz való kapcsolódás előtt érdemes ellenőrizni a biztonsági előírásokat és a megfelelő hozzáférési engedélyeket az adatbázisban, hogy megfelelően konfigurált és biztonságos kapcsolatot lehessen létrehozni.

9. ábra ChatGPT válasza a UiPath és Oracle adatbázis összekapcsolásához (Forrás: <https://chat.openai.com/c/11bb95d5-bb56-46bd-8ce7-51b242aa97a5>)

Ezek pedig:

- DWH-hoz csatlakozás
- DWH-ből való input fájl letöltés és kapcsolat zárás



10. ábra DWH kapcsolatot kiépítése (Forrás: Library_DWH\DWH_Connection\Estabilish_DWH_connection.xaml melléklet – saját közlés)

Először szintén a jogosultság adatait kell betöltenünk Orchestratorból, amit szintén paraméterként tudunk megadni. A Get Credential visszaad egy felhasználónevet mint string, és egy jelszót, mint SecureString. A kapcsolat kiépítésére a rendelkezésre álló database activity csomag, connect to database activity-jét használtam:

Connection String: "Dsn=DWHDATABASE;uid="+userId+";pwd="+pw

Provider Name: "System.Data.Odbc"

Ha a kapcsolat nem jönne létre ODBC-n keresztül, akkor megpróbálom OLEDB-n. Bizonyos esetekben előfordulhat, hogy nincs mind a két provider drivere telepítve az összes gépre, de az egyik biztosan elérhető.

Provider Name: "System.Data.OleDb"

Ez az activity pedig visszaad egy DatabaseConnection változót, amit lehet használni amíg szükségünk van az adatbázis kapcsolatra és szeretnénk írni/olvasni belőle. A folyamat végén pedig ezt majd szükséges lezárunk, de azt az activity-t nem szükséges könyvtárként létrehozni, mert csak egy bemeneti értéket vár, és az maga a kapcsolat.

A tábla ideiglenes fájlba való kimentésére pedig ugyancsak alapul szolgál a DWH kapcsolat kiépítése, majd a paraméterként megadható select query parancsot futtatja a már létrejött kapcsolaton. Paraméterként várja az activity még a létrehozandó xlsx Excel fájl teljes elérési útvonalát, a lapfűl nevét, valamint opcionális megadható egy update query is, ha szeretnénk az elhozott adatok státuszát változtatni a táblában.

Ha a létrehozandó fájl már létezik, akkor azt először töröljük, majd a select által kapott adattáblát Excel fájlba írjuk, és a rendeltetési helyére helyezzük. Ha sikeres volt a kimentés, az update query-t is futtatjuk, majd logoljuk a frissített sorok számát.

3.5. Robot kód fejlesztése

A továbbiakban a Performer robot "Framework\Process.xaml"-jét fejtem ki részletesen, hiszen tulajdonképpen ez az a kódrészlet, ami a valós tranzakcionális feladatvégrehajtást viszi véghez.

Először is a szükséges dependenciákat ellenőriztem, és használtam a lehető legújabb verziókat (2024.03.22), ezek az adatok megtalálhatóak a project.json fájlban.

```
"name": "Folyamatszervező_Riport_Keretrendszer_Performer",  
"projectId": "6a0698e7-d69b-47bd-ab75-e305d65b88ae",  
"description": "",  
"main": "Main.xaml",  
"dependencies": {  
  "DWH_Connector": "[1.0.1]",  
  "Microsoft.Activities": "[1.0.1]",  
  "SMTP_Connector": "[1.0.1]",  
  "UiPath.Excel.Activities": "[2.22.3]",  
  "UiPath.Mail.Activities": "[1.22.2]",  
  "UiPath.System.Activities": "[24.3.0]",  
  "UiPath.UIAutomation.Activities": "[23.10.11]",  
  "UiPath.WebAPI.Activities": "[1.18.0]"
```

Az SMTP_Connector és a DWH_Connector az előző fejezetben elkészített könyvtárak. Amire figyelni kellett, hogy ha a könyvtárak magasabb verziójú csomagokat használnak, előfordulhat kompatibilitási probléma; másik irányban nincs ilyen gon, hátrafelé kompatibilisek egymással a csomagok.

Ezen felül a project.json fájlban menthetjük el a robot folyamat nevét (name), és adhatunk neki leírást (description), ha szeretnénk.

Következő feladatunk a Config Excel fájl kitöltése minden olyan információval, amit majd a folyamatban, egy dictionary segítségével használni szeretnénk.

Name	Value	Description
OrchestratorQueueName	Folyamatszervező_riportok	Orchestrator Queue Name. Be sure to match this name with the one from the server. If there is anything passed in the Main as in OrchestratorQueueName argument, it overwrites this setting
Tower	Állománykezelés	Tower name of the business (optional)
Department	Folyamatszervezés	Department name of the business (optional)
Process	Folyamatszervezés_Riportok_Performer	Process name of the business (optional)
TransactionIDField	ID	Used for reporting queue and for logging. If empty, going to use the Reference as the Transaction ID Field.
Country	HU	Can be overwritten if the queue item contains the Country parameter.
MaxProcessTime	0	Max Process runtime in SECONDS. If 0 or not an Int32, the framework doesn't check the transaction runtime.

11. ábra Config Excel I. (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Data\Config.xlsx – saját közlés)

Name	Value	Description
MaxRetryNumber	0	Should be 0 if working with Orchestrator queues that have QueueRetry enabled. If > 0, the robot will retry the same transaction which failed with application exception. Must be integer
MaxConsecutiveRetryNumber	25	Number of continous application exceptions before the robot is stopped due to minimize faulty transactions
InitMaxRetryNumber	3	If > 0, Init could be retried by this many times in case of System Exception, otherwise it won't be retried. Must be integer
TimeoutShort	5000	Timeout short value in milliseconds, for activities which are likely to fail. Must be integer
TimeoutMedium	30000	Timeout medium value in milliseconds. Must be integer
TimeoutLong	120000	Timeout long value in milliseconds, for slow apps. Must be integer
ExScreenshotsFolderPath	\\HU0000005F001\L_havonta_torolt\RPA\Folyamatszervezés_Riportok_Performer	Where to save exceptions screenshots - can be a full or a relative path.
TempReportFolder	C:\Temp\TempReports	
DelayShort	1000	Delay short value in milliseconds, for activities where it is necessary to wait a little. Must be integer
DelayMedium	15000	Delay medium value in milliseconds, for activities where it is necessary to wait a moderate amount of time. Must be integer
DelayLong	60000	Delay long value in milliseconds, for activities where it is necessary to wait a long time. Must be integer
AccuracyLow	0.6	Image accuracy low value, for images that have high contrast. Must be double
AccuracyMedium	0.8	Image accuracy medium value, for images that have normal contrast. Must be double
AccuracyHigh	0.9	Image accuracy high value, for images that have low contrast. Must be double
LogMessage_InitError	Init Error:	
LogMessage_GetTransactionData	Processing Transaction Number:	Static part of logging message. Calling Get Transaction Data
LogMessage_GetTransactionDataError	Error getting transaction data for Transaction Number:	Static part of logging message. Error retrieving Transaction Data.
LogMessage_Success	Transaction Successful.	Static part of logging message. Processed Transaction succesful
LogMessage_BusinessRuleException	Business rule exception:	Static part of logging message. Processed Transaction failed with business exception
LogMessage_ApplicationException	System exception:	Static part of logging message. Processed Transaction failed with application exception
LogMessage_SuccessReason	Riport elkészült	Static part of logging message. Processed Transaction succesful with reason
SystemError_EmailTo	rpa_fpt@company.com	Email address to send alert messages
SystemError_EmailSubject	FATAL ERROR at #STEP with #PROCESS - #DATETIME	#PROCESS is replaced at runtime with Settings/ProcessName, #STEP with the process step (INIT, PROCESS, GETTRANSACTIONDATA), #DATETIME with current date + time
SystemError_EmailBody	Dear Automation Team There has been the following fatal error with the process #PROCESS at step #STEP: #MESSAGE at source #ERRORSOURCE. Please check the issue and restart the job if necessary. Kind regards, The #PROCESS bot	#PROCESS s is replaced at runtime with Settings/ProcessName #STEP with the process step (INIT, PROCESS, GETTRANSACTIONDATA) #DATETIME with current date + time, #MESSAGE with the exception message, #ERRORSOURCE with the exception source

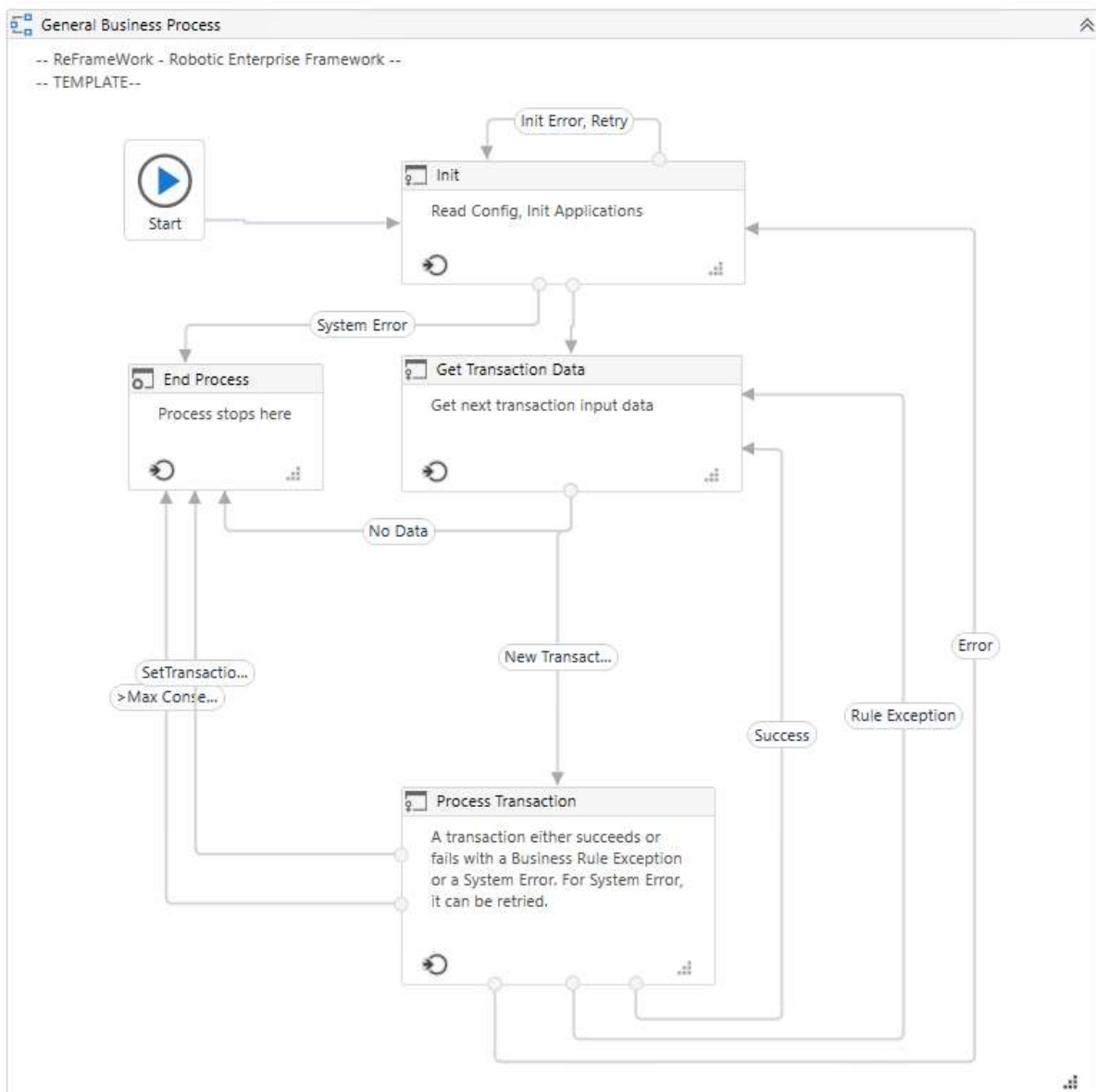
12. ábra Config Excel II. (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Data\Config.xlsx – saját közlés)

Az 5. ábrán a queue nevét adjuk meg, ahonnan a feldolgozandó tételeket (kimutatások, és azok paramétereiket) olvassa be a robot, valamint a folyamatról adhatunk meg adatokat, amiket például vizualizációhoz is fel tudunk használni később (országok, vagy osztályok/területek közti felbontás).

Valamint a max process time érték azt a cél szolgálja, hogy ha a megadott ideig (érték, másodpercben) nem fejeződik be egy tétel feldolgozása, akkor bármilyen beakadás esetén is képes legyen a robot továbbmenni a következő tranzakcióra.

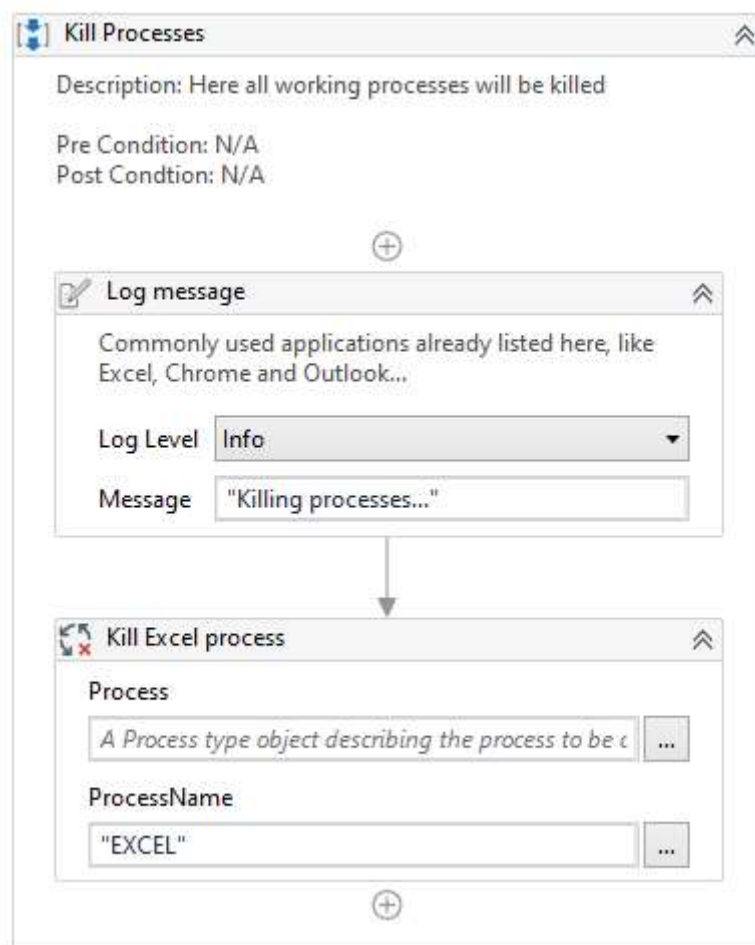
A 6. ábrán olyan általános adatokat adhatunk meg, hogy egy tételt hányszor próbálhat meg a robot újrafeldolgozni, a rendszereket hányszor próbálja megnyitni, vagy éppen hány folyamatos, egymás utáni rendszerhiba után kényszerítsük leállásra a robotot. Megadhatjuk még azt a mappát, ahová rendszerhiba esetén hova mentünk képernyőképet a hiba pillanatáról (ami segíti az üzletet a tétel kézi feldolgozásában, valamint a fejlesztőt is a hiba/hiányos specifikálás megoldásában). Valamint bármilyen egyéb, egyedi igényt kielégítő adatot felvihetünk.

Ezen a ponton elkezdhetjük feltölteni a QueueItem REFrameworkot (RE = Robotic Enterprise).



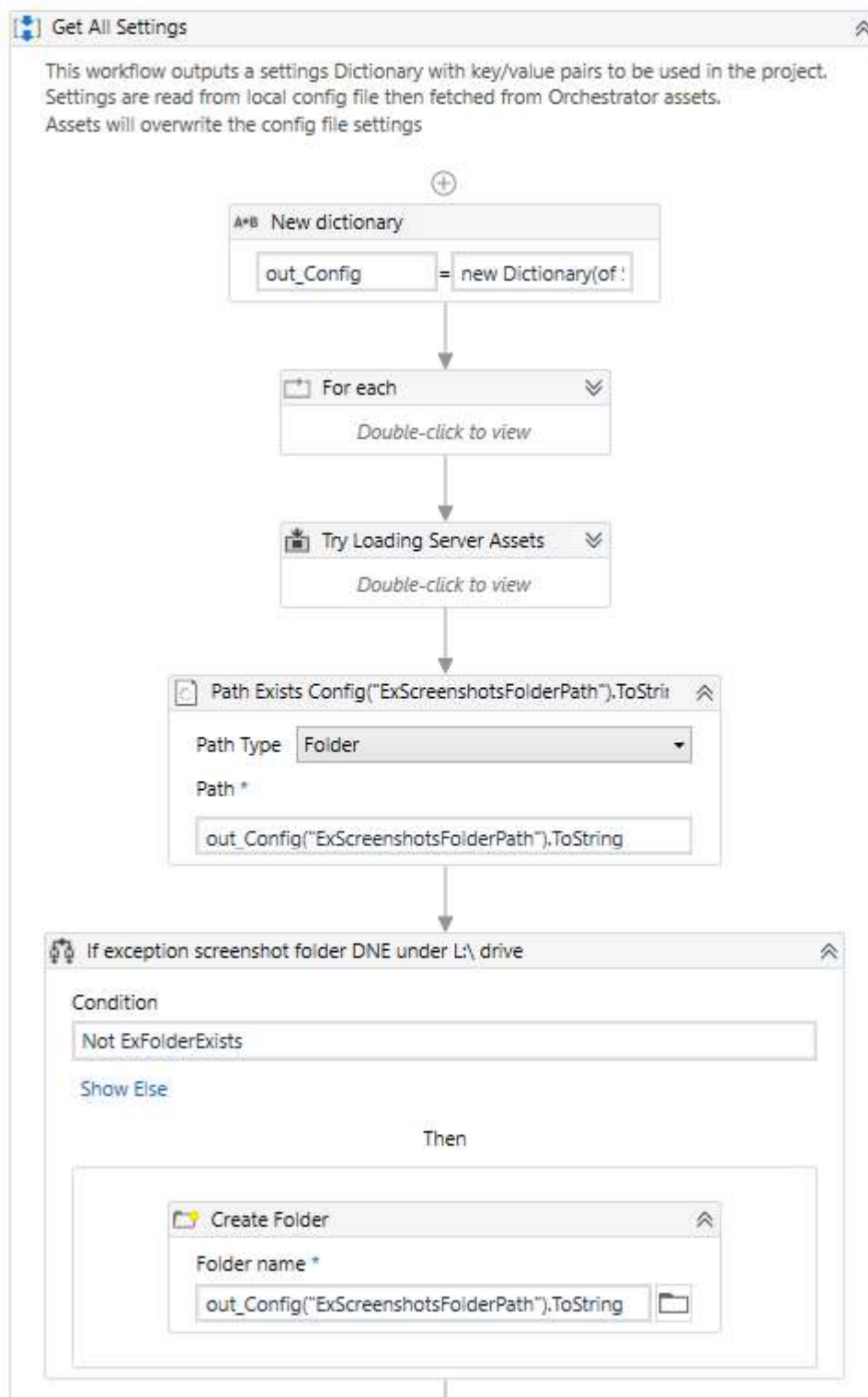
13. ábra REFramework state-machine ábrája ((Forrás: Folyamatszervezés riport
keretrendszer\Performer_DB.Activities\Main.xaml – saját közlés))

Az Init state-ben a Kill.xaml-ben csak az Excel.exe folyamatot termináljuk, mivel a folyamatunk későbbi részében használjuk az Excelt a riportok lementésére, és esetleges rendszerhiba esetén mindenképp vissza kell térnünk egy olyan állapotba, hogy semmilyen rendszer nincs beragadva.



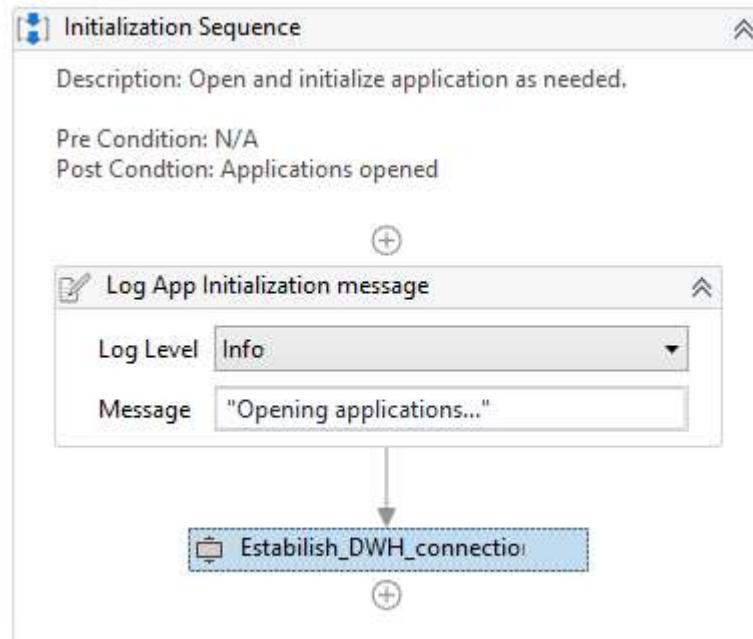
14. ábra Framework\KillAllProcesses.xaml (Forrás: Folyamatszervezés riport
keretrendszer\Performer_DB.Activities\Framework\KillAllProcesses.xaml melléklet – saját közlés)

Az InitAllSettings.xaml-ben a standard keretrendszer elemek mellett (ami elkészíti a config dictionary-t a config Excelből) a robot ellenőrzi, hogy létezik-e az a mappa, ahova a hibaesetekről készülő képernyőképeket kell menteni. Ha még nem létezik (első futás adott gépen esetében, vagy manuális mappatörlés esetében) létrehozza az adott mappát (itt előzetesen szintén jogosultságigényt kell igényelni a célmappához). Ezt már megtettem korábban, és minden esetben egy mappa alá készítünk almappákat, amihez a jogosultságot a korábban említett munkaköri sablon tartalmazza). Az InitAllSettings csak akkor van meghívva, hogy ha a config dictionary még nem létezik; ha többszöri újrapróbálkozás meg van engedve az Init state-nek, akkor csak első alkalommal hívódik meg.



15. ábra Framework\InitAllSettings.xaml (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Framework\InitAllSettings.xaml melléklet – saját közlés)

InitAllApplications.xaml-ben a korábban létrehozott DWH kapcsolat létrehozása activity szerepel csak. Ennek 2 paramétere van, egy in és egy out. Az activity elvárja a DWH hozzáférés asset nevét, amiből magának lehívja a felhasználónevet és jelszót, amit szükséges átadni a DWH adatbáziskapcsolathoz.



Properties

General_PLSQL.DWH_Connection.Estabilish_DWH_connection

Common

DisplayName: Estabilish_DWH_connection

Input

in_DWH_Credential: "PLSQL_Credential"

Misc

Private: ☐

Output

out_DBConnection: out_DBConnection

16. ábra Framework\InitAllApplications.xaml (Forrás: Folyamatszervezés riport
keretrendszer\Performer_DB.Activities\Framework\InitAllApplications.xaml melléklet – saját közlés)

A tényleges hozzáférési adatok az Orchestrator-ban találhatóak, és tudjuk beállítani az asset-hez tartozó felhasználónevet és jelszót. Ez lehetővé teszi, hogy a kód változtatása nélkül képesek legyünk a jelszót frissíteni például, ha nincs rá lehetőség, hogy technikai felhasználóként, nem lejárt jelszó jogosultságok kapjon a robot.

Development > Assets > Edit Asset: PLSQL_Credential

General details

Asset name *

PLSQL_Credential

Type

Credential

Description

Credential used for DWH connection

Credential Store

Orchestrator Database

Tags

Labels

DWH

Properties (Key-value pairs)

Add new

This object contains no properties.

Asset value

Global Value

Username *

x606015

Password *

The username has changed, please set a new password.

Add robot asset value

Robot	Value
No data to display yet.	

Cancel
Update

17. ábra Hozzáférési adatok menedzselése az Orchestrator-ban (Forrás: saját közlés)

A Get Transaction Data state-ben IT infrastrukturális megbízhatatlanság miatt további hibakezeléseket kellett beépíteni. A keretrendszer csak egy Orchestrator API segítségével kért le egy QueueItem változót az Orchestrator megadott queue-jából, viszont ez az API időközönként időtúllépés miatt hibára futott, így Nothing értéket kapott a TransactionItem változó, és a robot az End state-be került. Ugyanakkor az Orchestrator bizonyos esetben mégis úgy jelezte, hogy le lett kérve tétel, és a státuszát már „in progress”-é változtatta, de a hiba miatt ez a tétel csak lebegett a semmiben.

Ezt kiküszöbölve az alábbi hibakezelést építettem be:

- A Get Transaction Item activity-t egy Try-Catch keretbe tettem, ez került a Try részbe
- Hiba esetén az összes in-progress QueueItem-et lekérem egy IEnumerable<QueueItem> változóba
- Ha az in-progress tételek száma 0, akkor csak újrageneráljuk hibát, és a retry-scope újrapróbálja a lekérést
- Ha legalább 1 in-progress QueueItem van, az összes ilyen tétel esetében:

- Meghatározom a gép nevét a Environment.MachineName segítségével, és ennek a segítségével megkapom a robot Orchestrator-ban tárolt nevét a `"/odata/Robots?$filter=MachineName%20eq%20"+machineName+""` kérés révén
- ezután megvizsgálom az éppen aktuális in-progress queue item robot gazdáját: `"/odata/QueueItems("+queueitem.QueueDefinitionId.ToString+")?$expand=Robot"`
- ha a robotunk azonos a QueueItem robot gazdájával, akkor kilépünk a vizsgálati ciklusból, és ezt a tételt vesszük fel mint TransactionItem
- ha nincs olyan QueueItem, amit az adott robot kért volna már le, akkor is újrageneráljuk a hibát, és a retry-scope újrapróbálkozik
- Ezt az egészet pedig egy retry-scope helyeztem, 10 próbálkozással, próbálkozások között 1 másodperc várakozással

Ezen felül szükségünk van még egy időbélyegzőre, amikor is elkezdődött a feldolgozása a tételnek (TransactionStartTime). Ezt fogjuk használni amikor a feldolgozás eredményét státuszoljuk, valamint kimutatásokhoz is használhatjuk (ciklusidő, valamint robot erőforrás számításához például).

A robot fő része pedig a Process.xaml-ben került meghatározásra.

Ellenőrizzük, hogy a config-ban rögzített TempReportFolder mappa létezik-e; ha létezik töröljük, majd újra létrehozzuk. Ez a lépés azért szükséges, hogy hiba esetén a korábban lementett kimutatásokat biztosan töröljük és ne okozunk semmilyen összeakadást (akár újrapróbálkozás esetén se).

A robot elvégzi a leggyakoribb adat szintű validációkat, ami kiszűri az adott esetben hibásan átadott kimutatás igényeket:

- Ha a TRANSFER_TYPE = „NETWORK” és a FOLDER üres → Business Rule Exceptiont generálunk: „Network esetén Folder megadása kötelező”

Majd ugyan ez fordított logikával:

- Ha a TRANSFER_TYPE = „EMAIL” és az ADDRESS üres → Business Rule Exceptiont generálunk: "Email esetén Address megadása kötelező"

Ezután egy 10 ismétléses for ciklust használunk, a maximalizált 10 kimutatás lementésére (ForEach report in Enumerable.Range(0,10).ToArray())

A könnyebb kezelhetőség és átláthatóság miatt helyi változókat használunk:

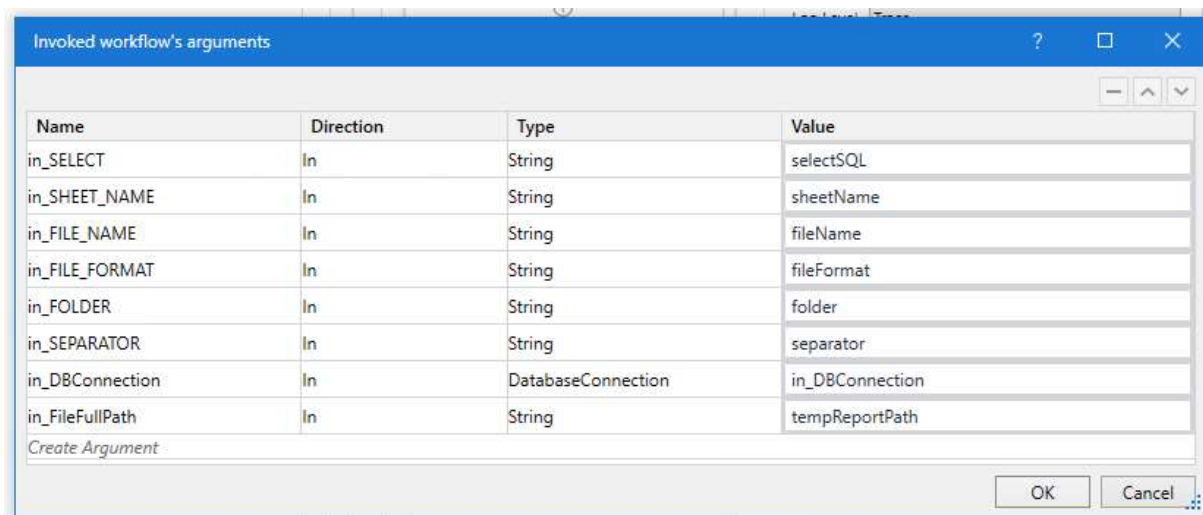
- **folder** = in_TransactionItem.SpecificContent("FOLDER").ToString
- **selectSQL** =
in_TransactionItem.SpecificContent("SELECT_" + (n+1).ToString("00")).ToString
- **sheetName** =
in_TransactionItem.SpecificContent("SHEET_NAME_" + (n+1).ToString("00")).ToString
- **fileName** = in_TransactionItem.SpecificContent("FILE_NAME").ToString
- **fileFormat** = in_TransactionItem.SpecificContent("FILE_FORMAT").ToString
- **separator** =
in_TransactionItem.SpecificContent("FILE_CSV_SEPARATOR").ToString

Ha a folder változó üres (azt jelenti, hogy emailen keresztül kell kiküldeni a riportot), a tempReportPath változó a in_Config("TempReportFolder").ToString + "\" + fileName + "." + fileFormat értéket kapja meg

Ha pedig nem üres a folder változó, akkor szintén létrehozuk a tempReportPath változót, ugyanezzel az értékkel, valamint a fullReportPath változót, folder + "\" + fileName + "." + fileFormat értékkel.

Ezután vizsgáljuk a selectSQL értékét; ha üres, akkor kilépünk a for ciklusból, mert az összes kimutatás elkészült (és feltételezzük, hogy az üzlet nem hagy ki index pozíciókat az átadott input adatban).

Ha a selectSQL nem üres, meghívjuk a "TableDownload.xaml" kódot.



18. ábra TableDownload.xaml bemenő paraméterei (Forrás: Folyamatszervezés riport
keretrendszer\Performer_DB.Activities\TableDownload.xaml melléklet – saját közlés)

A TableDownload végrehajtja az SQL select parancsot az Init-ben létrehozott adatbáziskapcsolat felhasználásával, majd a fileFormat alapján lementi vagy Excel fájlba, vagy csv-be.

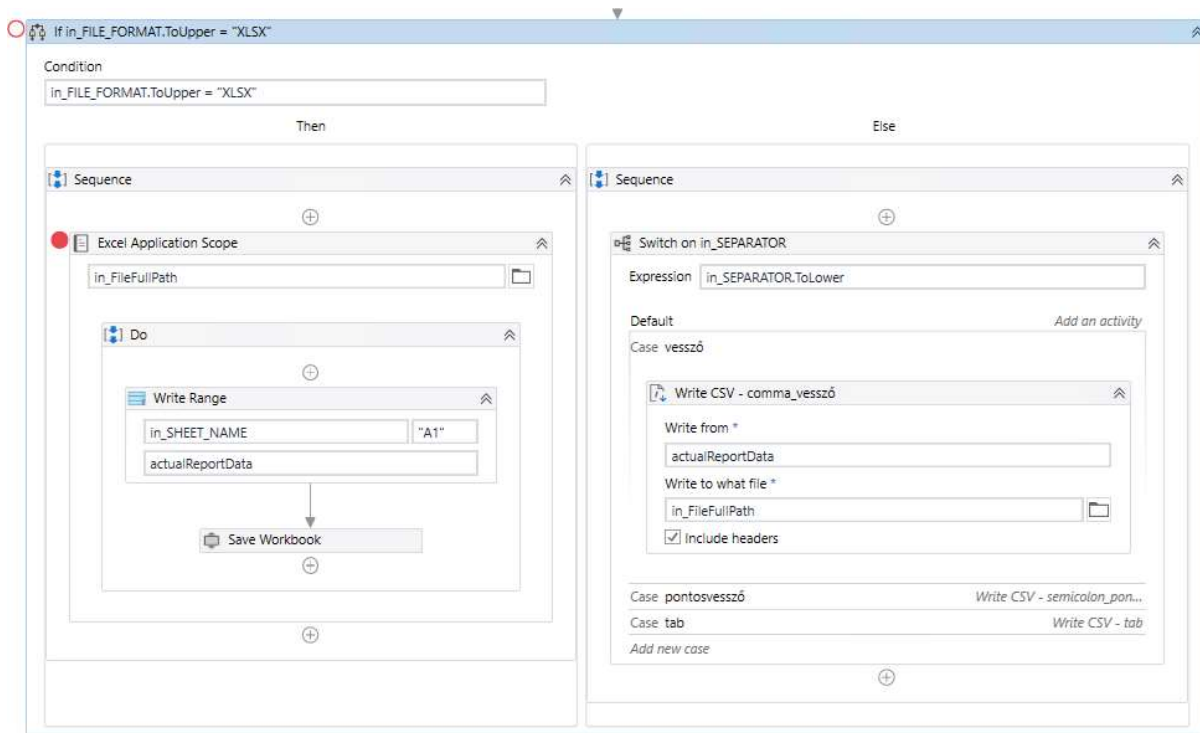
Excel fájl esetében kénytelenek voltunk Excel Application Scope²-ot használni, ami a gépre telepített Excelt használva hozza létre az állományt, mert a Workbook activities write range activity-je nagy fájlok létrehozása esetén (~25 MB felett) terminálta memóriakezelési probléma miatt a robotot 0xc0000005 hibával. Az Excel Application Scope kevesebb erőforrást használ, és 500 MB-ig tesztelve hiba nélkül működött; az üzleti elvárás ugyan nem definiálta előre az átlagos fájl méretet, de 50 MB-nál nagyobb méretű állományokat nem állítanak elő.

Mivel a lapfűl neve is megadható a select SQL mellett, így a robot képes 1 Excel fájlba írni az adatokat, a megadott fülekre. Szintén feltételezés, hogy a lapfűlek nevei egyediak lesznek tételen belül.

CSV fileFormat esetében pedig a megadott szeparátor használatával textfájlba írjuk a kapott adattáblát.

Mindkét esetben az AddHeaders property True-ként van megadva, mert szükség van a fejlécre a kimutatásokban.

² <https://forum.uipath.com/t/difference-between-excel-and-workbook-activities/4025>



19. ábra TableDownload.xaml működése (Forrás: Folyamatszervezés riport
keretrendszer\Performer_DB.Activities\TableDownload.xaml melléklet – saját közlés)

Excel fájlok esetében továbbá a robot elvégzi a formai és titkosítási beállításokat. Az RPA eszköz natívan nem támogatja az ilyen mértékű Excel interakciót, úgyhogy itt 2 különböző VBA makrót hív meg a robot, amit külön készítettem el. Formázás esetében:

Entry	method	parameters:
{in_TransactionItem.SpecificContent("DATATYPE_TEXT").ToString,		
in_TransactionItem.SpecificContent("AUT_COLUMN").ToString,		
in_TransactionItem.SpecificContent("FIX_ROW").ToString,		
in_TransactionItem.SpecificContent("FIX_COLUMN").ToString,		
in_TransactionItem.SpecificContent("AUT_FILTER").ToString,		
in_TransactionItem.SpecificContent("ZOOM").ToString,		
in_TransactionItem.SpecificContent("FIRST_PAGE").ToString}		

```

Sub formatting(text_data, autowidth, fixrow, fixcol, ApplyFilter, ZoomLevel, lapfűlszámaforopen)
Dim WS As Worksheet
For Each WS In ActiveWorkbook.Worksheets
    'Activate each sheet
    WS.Activate
    'Set zoom level
    ActiveWindow.Zoom = CInt(ZoomLevel)
    'Set A1 as default Cell
    Cells(1, 1).Select
    'Set text formatting
    If text_data= "IGEN" Then
        Range(Selection, ActiveCell.SpecialCells(xlLastCell)).Select
        Selection.NumberFormat = "@"
        Cells(1, 1).Select
    End If
    'Apply auto filer
    If ApplyFilter = "IGEN" Then
        WS.Range("B1").AutoFilter
    End If
    'Apply auto column
    If autowidth = "IGEN" Then
        WS.Columns.AutoFit
    End If
    'Freeze panels row and/or col
    With ActiveWindow
        setPane = False
        If .FreezePanes Then .FreezePanes = False
        If CInt(fixrow) > 0 Then
            .SplitRow = fixrow
            setPane = True
        End If
        If CInt(fixcol) > 0 Then
            .SplitColumn = fixcol
            setPane = True
        End If
        If setPane = True Then
            .FreezePanes = True
        End If
    End With
    'Delete column1 (plsql hiba)
    If Cells(1, 1) = "Column1" Then
        Columns("A:A").Select
        Selection.Delete Shift:=xlToLeft
        Range("A1").Select
    End If
Next WS
'Activate main sheet
Worksheets(CInt(lapfűlszámaforopen)).Activate
End Sub

```

Valamint titkosítás esetében:

Entry	method	parameters:
{in_TransactionItem.SpecificContent("FILE_ENCRYPTION").ToString,in_TransactionItem.SpecificContent("FILE_PASSWORD").ToString,in_TransactionItem.SpecificContent("READING_WITHOUTPASSWORD").ToString,in_TransactionItem.SpecificContent("READING_ONLYWITHPASSWORD").ToString}		

```

Sub titkositas(encr, pw, withoutpw, onlywithpw)
Application.UserName = "VGTF"
ThisWorkbook.BuiltinDocumentProperties("Author") = "VGTF"
Application.DisplayAlerts = False
With ThisWorkbook
    'Lapvédelem w/o password
    If encr = "IGEN" Then
        ActiveWorkbook.Protect
        ActiveWorkbook.Save
    End If
    'Jelszó védelem
    If withoutpw = "IGEN" And onlywithpw = "IGEN" Then 'Megnyitásra és editálásra
        .SaveAs Filename:=ThisWorkbook.Path & "\" & ThisWorkbook.Name, CreateBackup:=False, Password:=pw, WriteResPassword:=pw
    ElseIf withoutpw = "NEM" And onlywithpw = "IGEN" Then 'ReadOnly ok
        .SaveAs Filename:=ThisWorkbook.Path & "\" & ThisWorkbook.Name, CreateBackup:=False, Password:=pw
    ElseIf withoutpw = "IGEN" And onlywithpw = "NEM" Then 'Password mandatory
        .SaveAs Filename:=ThisWorkbook.Path & "\" & ThisWorkbook.Name, CreateBackup:=False, WriteResPassword:=pw
    Else
        'Do Nothing
    End If
End With
Application.DisplayAlerts = True

Set WSHnet = CreateObject("WScript.Network")
UserName = WSHnet.UserName
UserDomain = WSHnet.UserDomain
Set objUser = GetObject("WinNT://" & UserDomain & "/" & UserName & ",user")
UserFullName = objUser.FullName
Application.UserName = UserFullName & " / " & Environ("username")
ThisWorkbook.BuiltinDocumentProperties("Author") = UserFullName & " / " & Environ("username")
End Sub

```

Ezután a robot már csak a TRANSFER_TYPE alapján elvégzi az utolsó lépést; hálózati meghajtóra ment vagy e-mailt küld.

NETWORK érték esetén csak áthelyezi a tempReportPath állományt a fullReportPath állományba (mindkét változót korábban létrehozta és meghatározta a teljes elérési útvonalukat, fájlnevvvel, kiterjesztéssel).

EMAIL esetén pedig ellenőrzi, hogy létrehozott állomány mérete nem haladja-e meg a FILE_SIZE_LIMIT-ben meghatározott maximális fájl méretet (ha nincs megadott limit, akkor bármekkora fájl kiküldhet a robot).

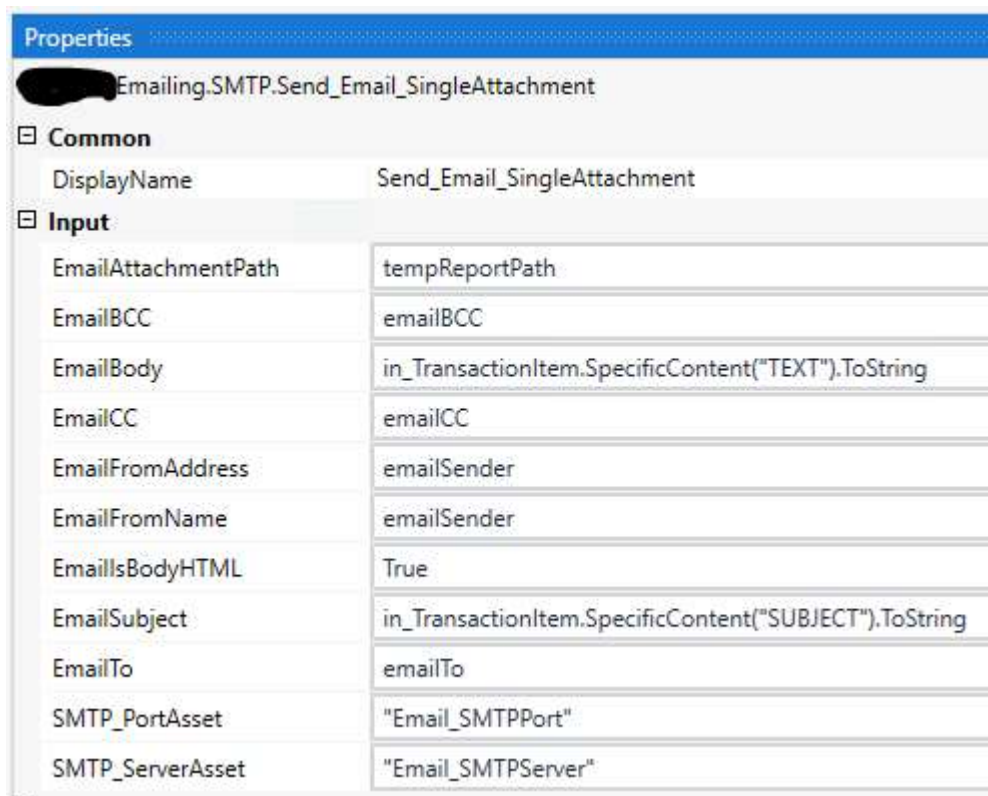
```

fileMéret = New System.IO.FileInfo(tempReportPath).Length
fileMéret = fileMéret/1024/1024
fileMéretLimit = in TransactionItem.SpecificContent("FILE_SIZE_LIMIT").ToString
If String.IsNullOrEmpty(fileMéretLimit) Then
    fileMéretLimit = (fileMéret + 1).ToString
End If

```

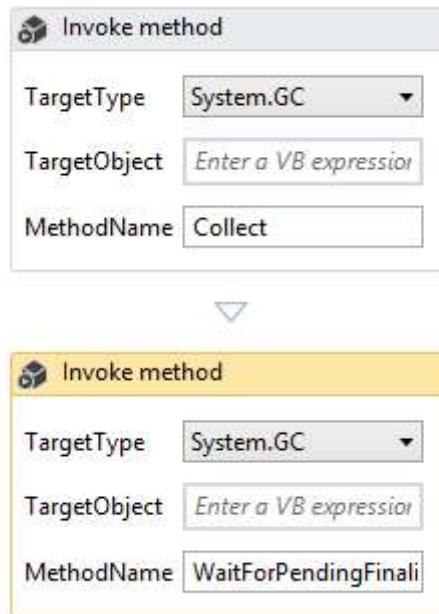
Így ha a **fileMéret** > Cdbl(**fileMéretLimit**), akkor üzleti kivételt generál szintén a robot: New Business Rule Exception("A megadott fájl már nem küldhető ki a mérete miatt").

Végezetük SMTP-n elküldi az e-mailt a robot a címzekketnek a paraméterek alapján:



20. ábra SMTP küldés paramétereit (Forrás: Library_Emailing\SMTP\Send_Email_SingleAttachment.xaml melléklet – saját közlés)

Email küldés végeztével a robot törli a tempReportPath állományt. Hogy kiküszöböljük azt a hibát, amikor az email küldés használja még a csatolmányt és nem engedi a Windows törölni a fájlt, a garbage collector segítségével felszabadítjuk a már nem használt memóriát, a System.GC Collect és System.GC WaitForPendingFinalizers metódusokkal.



21. ábra Garbage Collector Collect és WaitForPendingFinalizers metódusok (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Process.xaml melléklet – saját közlés)

A Process state-ben, a Process.xaml elvégzése után visszaírjuk az input táblába a feldolgozás eredményét, szintén a DWH adatbáziskapcsolat használatával:

"UPDATE

" + TransactionItem.SpecificContent("DWH_Source").ToString + ".WRK_RPA_EXCEL_EMAIL SET RPA_STATUS = '" + Status + "'", RPA_ERCODE = '" + Reason + "'", RPA_START = '" + TransactionStartTime.ToString("yyyyMMddHHmmss") + "', RPA_END = TO_CHAR(CURRENT_TIMESTAMP, 'YYYYMMDDHH24MISS') WHERE ID = '" + TransactionItem.SpecificContent("ID").ToString + "'"

Végül az End Process state-ben, a Framework\CloseAllApplications.xaml-ben lekapcsolódunk a DWH adatbáziskapcsolatról.

3.6. Queue-k létrehozása

Ehhez a folyamathoz csak egyetlen queue-ra van szükségünk. Auto retry-t nem használunk, mert a folyamat adatvezérelt, ha elsőre nem sikerül a tétel feldolgozása, akkor másodjára se fog sikerülni, mert a folyamatban nincs semmilyen GUI ami képes lenne váratlan hibát generálni.

Szintén nem használjuk a Enforce unique references lehetőséget, mert a QueueItem-ek reference értéke az input adattábla egyedi ID-ja lesz minden esetben.

Valamint ebben az esetben a Store in Encrypted format lehetőséget se kell választanunk, mert nem szerepel érzékeny adat a folyamatokban (ügyfeladatok esetében ez a jelölőnégyzet kötelezően bepipálandó a saját folyamatunk szerint).

Development > Queues > Create Queue

General details

Name *

Folyamatszervező_riportok

Description

Queue used for reporting activities

Additional options

☐ Enforce unique references

☐ Store in encrypted format ⓘ

☐ Auto Retry

Tags

Labels

DWH ×

Properties (Key-value pairs) [Add new](#)

This object contains no properties.

Schema Definitions

Specific Data JSON Schema [Browse](#)

Output Data JSON Schema [Browse](#)

Analytics Data JSON Schema [Browse](#)

SLA Predictions

☐ Enable SLA for this queue

[Cancel](#) [Add](#)

22. ábra Queue létrehozása Orchestrator-ban (Forrás: saját közlés)

3.7. Asset-ek létrehozása

A Robot kód fejlesztése részben már létrehoztuk a credential típusú asset-ünket, és ezen felül csak egy email disztribúciós listát szeretnénk Orchestrator asset-ben tárolni, hogy ha az input tábla üres napi futás esetén, akkor dinamikusan változtatható legyen a címzettek köre. Ez egy egyszerű text típusú asset.

3.8. Robot kód üzleti tesztelése

Az üzleti teszt esetében a követelményspecifikáció minden egyes pontjára dedikált teszt esetet állítottunk elő az üzleti területtel.

- Egy tábla, xlsx formátumban
- Több tábla, xlsx formátumban
- 10 tábla, xlsx formátumban
- Egy tábla, csv formátumban

- CSV lementése vessző elválasztóval
- CSV lementése pontosvessző elválasztóval
- CSV lementése tab elválasztóval
- Megadott nevű, megadott lapfűlnevű xlsx kimutatás hálózati meghajtó megadott mappájában előállt
- Megadott nevű, csv kimutatás hálózati meghajtó megadott mappájában előállt
- Megadott nevű, megadott lapfűlnevű xlsx kimutatás emailen elküldve
- Megadott nevű, csv kimutatás emailen elküldve
- xlsx kimutatás lapfűlvédelemmel
- xlsx kimutatás olvasásra megnyitható jelszó nélkül opcióval
- xlsx kimutatás olvasásra megnyitható jelszóval opcióval
- xlsx kimutatás olvasásra megnyitható jelszó nélkül és olvasásra megnyitható jelszóval opcióval
- xlsx kimutatás szövegre állított formázással
- xlsx kimutatás automatikus oszlopszélességgel
- xlsx kimutatás rögzített oszloppal
- xlsx kimutatás rögzített sorral
- xlsx kimutatás autofilter bekapcsolásával
- xlsx kimutatás nagyítás beállításával
- xlsx kimutatás megadott lapfűlrel aktiválva megnyitáskor
- Email küldés esetén meghatározott fájl méret feletti kimutatás
- Email jelzés, ha nincs új feldolgozható kimutatás a táblában

Ezen felül a tesztadat előállítás során manuális adathibák miatt az alábbi esetek lettek, véletlenszerűen, tesztelve:

- Több tábla, CSV formátumban → hiba ugyan nem történt, de a robot felülírta az adatállományt a legutolsó SQL select tartalmával
- NETWORK transfer type esetén hiányzó folder érték → beépítettük a hibakezelésbe
- EMAIL transfer type esetén hiányzó address érték → beépítettük a hibakezelésbe
- Hiányzó táblajog az SQL select-ben → ezt a framework kezeli le, mint rendszerhiba. Előzetesen nem tudjuk validálni, csak a lekérdezés pillanatában.
- Több tábla ugyan olyan névvel való lementése → ez tesztadat hibaként azonosítottuk, nem került beépítésre, éles működés közben ilyen nem lehet az előállító procedura szerint

3.9. Robot élesítése

Amennyiben robotunk készen áll az élesítésre, publikáljuk éles környezetbe a kódot és élesítjük az új robot folyamatot, valamint beállítjuk az ütemezést. Az élesítés folyamán a létrehozott asset-eket és queue-kat is át kell másolnunk az éles környezetbe az Orchestrator felületén.

A Studio-ból történő élesítéskor az alábbiakat állítjuk be:

- Package name: a korábban a project.json fájlban már beállítottuk, de itt is tudnánk még módosítani
- New version: megadjuk milyen verziót szeretnénk használni, de hagyhatjuk, hogy a rendszer automatikusan adjon új verziószámot.
- Is prerelease: nem kattintjuk be a jelölőnégyzetet. Akkor használjuk, ha élesítés előtti tesztelést végzünk, teljesen szimulálva az éles környezeti működést.
- Release Notes: Rövid üzenetben megadjuk miről szól a publish, milyen változtatás történik benne. Szerepel benne még egy JIRA Epic szám ami az igényhez kapcsolódik, vagy hibajegyzék szám éles hibabejelentés esetén.

The screenshot shows the 'Publish Process' window in the Studio application. The window has a blue title bar with the 'Ui' logo and the text 'Publish Process'. On the left is a sidebar with three tabs: 'Package properties' (selected), 'Publish options', and 'Certificate signing'. The main content area is titled 'Package properties' and contains several form fields: 'Package Name *' with a dropdown menu showing 'Demo_Process'; 'Version' section with 'Current Version' (1.0.0) and 'New Version *' (1.0.1) text boxes; an unchecked checkbox for 'Is Prerelease ?'; 'Package Icon ?' with a file selection button and the text 'Optional Package Icon'; 'Project tags ?' with a search bar containing the placeholder text 'Start typing to get a list of possible matches'; and 'Release Notes' with a text area containing 'Élesítés. JIRA Epic XX-123.'. At the bottom of the window are four buttons: 'Cancel', 'Back', 'Next', and 'Publish'.

23. ábra Publish a Studio-ból (Forrás: saját közlés)

Publish után az Orchestrator-ba lépve az új process-t is élesíteni kell. Ez az Automations/Processes alatt tesszük meg, majd az Add Process gombot választjuk.

A package source name legördülőből kiválasztjuk a folyamatunkat. A többi értéket hagyhatjuk alapértelmezetten.

Az additional settings alatt megadhatjuk a azt a nevet, ahogy az Orchestrator-ban szeretnénk látni; ha üresen hagyjuk akkor a project.json-ben megadott értéket láthatjuk itt is. A korábban megadott descriptiont itt felülírhatjuk, vagy ha eddig nem adtunk leírást megtehetjük, de nem kötelezően töltendő a mező.

A job priority-t hagyjuk jelen esetben mediumon, ugyanakkor, ha kritikus folyamatról van szó, ezt egy 10 osztásos skálán állíthatjuk. Ennek akkor van jelentősége, ha több folyamán várakozik szabad erőforrásra, akkor a magasabb prioritású indul előbb, egyébként a FIFO (first in, first out) elv érvényesül.

Majd a create gombra nyomva, a robotikus folyamatunk már futtatható, vagy ütemezhető.

Az ütemezést az Automations/Triggers menüpont alatt tesszük meg az add new trigger gomb megnyomásával. Kiválasztjuk a process nevét, amit az előbb élesztettünk, ha szükséges itt is módosíthatunk a prioritáson, vagy hagyjuk öröklődni a processből. Adunk az ütemezésnek is egy nevet, meghatározzuk hány és melyik gépen fusson a folyamatunk, majd megadjuk az ütemezési adatokat. A felület a leggyakrabban használt ütemezésre ad lehetőséget (minden percben, órában, napban, a hét megadott napján... stb), vagy használhatunk CRON ³kifejezést is az ütemezés megadására. Üzemeltetést segítve beállíthatjuk egy-egy folyamat mikor álljon le, még akkor is ha lenne feldolgozandó tétel, hogy az erőforrásokat a leghatékosabban ki tudjuk használni. Végezetül az add gombra kattintva a robotunk a következő ütemezési időpontban el fog indulni a megadott erőforrás(ok)on.

³ általam használt CRON készítő: <https://www.freeformatter.com/cron-expression-generator-quartz.html>

Development > Automations > Triggers > Create Trigger

Time
Schedule a recurrent time to start a job.

Queue
Start a job when new items are added to a queue.

Name *

Timezone *

Process Name *

Job priority *

Execution Target
☒ All Robots
☐ Specific Robots
☐ Dynamic Allocation

Arguments

☐ Minutes
☐ Hourly
☐ Daily
☐ Weekly
☐ Monthly
☒ Advanced

Cron expression 0 0 10 15W * ? *
The process will be scheduled in (UTC+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague and automatically adjusted for daylight saving time.

Non-working days restrictions

☒ Schedule ending of job execution
 a job after days hours mins

24. ábra Ütemezés az Orchestrator-ban (Forrás: saját közlés)

3.10. Robot üzemeltetése

Hogy biztosítsam, hogy a folyamat képes legyen minden rendelkezésre álló gépen ugyan úgy lefutni, létrehoztam egy rendszer ellenőrző robotot, amit minden nap, minden gépen lefut. Ez az ellenőrzés nem csak a szakdolgozatban használt folyamatban használt rendszerek ellenőrzésére szolgál, hanem az összes, már létező robotikus folyamat által használt rendszerek beállításának ellenőrzésére is szolgál.

Ez az alábbiakat ellenőrzi:

- Google Chrome UiPath add-in elérhető, és a robot képes dolgozni a böngészővel
 - Céges intraneten Chrome böngészőt használva, ha egy stabilnak vélt selector-t megtalál a robot, akkor az add-in működik
- Képernyőfelbontás ellenőrzése
 - **out_resolution** =
System.Windows.Forms.Screen.PrimaryScreen.Bounds.Width.ToString + " x "
+ System.Windows.Forms.Screen.PrimaryScreen.Bounds.Height.ToString
- C:\ meghajtó méretének és szabad tárterületének ellenőrzése

```

DriveInfo = New DriveInfo(in_Drivename)
SpaceAvailable = DriveInfo.AvailableFreeSpace
out_FreeDiskSpace = math.Round((SpaceAvailable / (1024 * 1024*1024)),2).ToString
SpaceTotal = DriveInfo.TotalSize
out_TotalDiskSpace = math.Round((SpaceTotal / (1024 * 1024*1024)),2).ToString

```

3.11. Adatvizualizáció, riportok, visszamérés

Vizualizáció_log.xlsx csatolmány.

Még korábban említettem, hogy az RPA képes időt és erőforrást felszabadítani, de mérések nélkül ez is csak egy bölcs rabbi gondolat. Mind az üzleti, mind a fejlesztői oldal, valamint a menedzsment és szponzor számára is fontos miként járulnak hozzá a robotok a vállalat eredményességéhez.

Ezekhez a mérésekhez az alábbi információt használjuk fel, és készítünk belőle riportot:

- **Process:** a robotikus folyamat neve
- **Department:** terület/osztály a vállalaton belül. Az adatokat készíthetjük terület szinten, vagy folyamat szinten. A szakdolgozatban a további részekben folyamat szinten készítünk kimutatást.
- **Cycle Time:** a robot által elvégzett tételek ciklusideje
- **Status:** a kimenetel státusza. Megkülönböztetjük a sikeres, sikertelen, valamint üzleti kivételeket
- **Reason:** a státuszhoz kapcsolódó ok, kvázi alstátusz. Üzleti kivétel esetében a hibaüzenet maga, sikeres tételeknél pedig megkülönböztethetjük a különböző sikeres kimeneteket egymástól amennyiben a folyamat igényli.
- **TimeStamp:** a tétel feldolgozásának befejezésének az időpontja

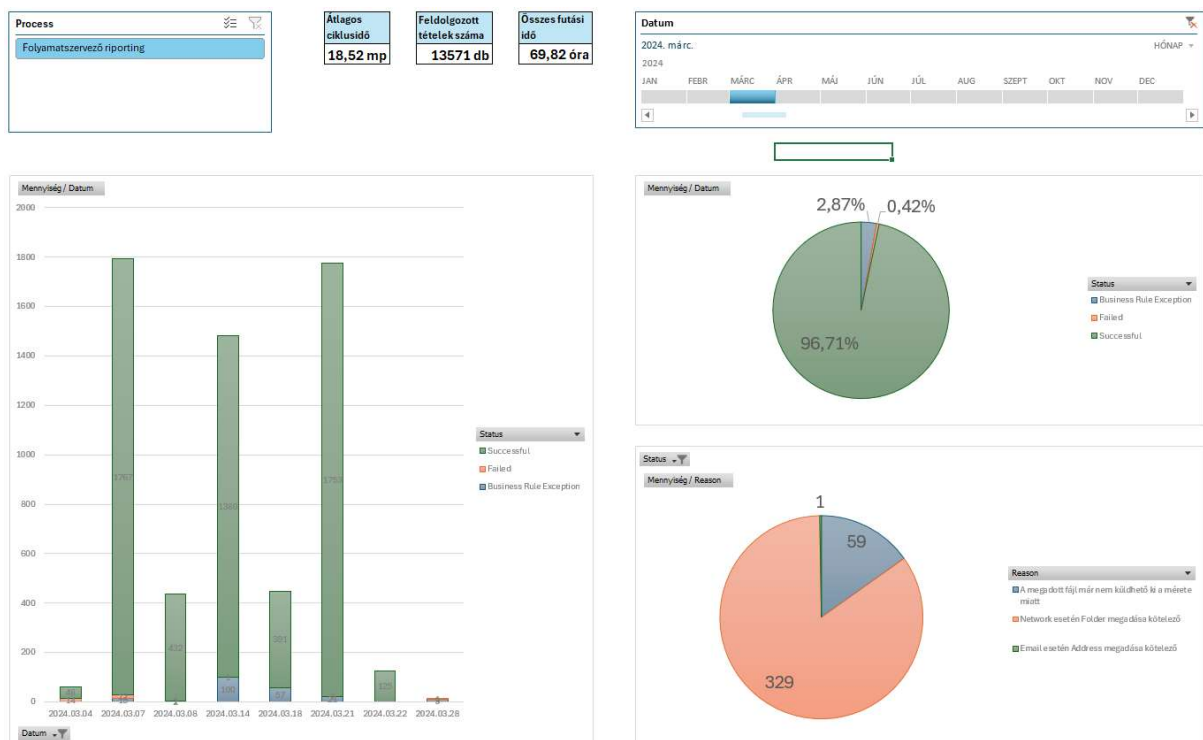
Process	Department	Cycle Time	Status	Reason	TimeStamp	Datum	Reference
Folyamatszervező riporting	Reporting and analytics	39	Successful	NETWORK report elkészült	2024.03.14 12:34:42	2024.03.14	TKT3150019
Folyamatszervező riporting	Reporting and analytics	37	Successful	NETWORK report elkészült	2024.03.14 12:42:16	2024.03.14	TKT3083783
Folyamatszervező riporting	Reporting and analytics	36	Successful	NETWORK report elkészült	2024.03.07 13:06:51	2024.03.07	TKT3137488
Folyamatszervező riporting	Reporting and analytics	35	Successful	NETWORK report elkészült	2024.03.04 14:12:41	2024.03.04	TKT2967780
Folyamatszervező riporting	Reporting and analytics	35	Successful	NETWORK report elkészült	2024.03.07 11:33:26	2024.03.07	TKT3133788
Folyamatszervező riporting	Reporting and analytics	35	Successful	NETWORK report elkészült	2024.03.07 13:34:12	2024.03.07	TKT3135499
Folyamatszervező riporting	Reporting and analytics	34	Successful	NETWORK report elkészült	2024.03.14 12:37:42	2024.03.14	TKT3151343

25. ábra Az adatvizualizáció adatvagyona (Forrás: Vizualizáció_log.xlsx melléklet – saját közlés)

Ezeket az információkat a feldolgozás során gyűjti össze a robot a config Excel, valamint a futás alatt kikalkulált adatok felhasználásával (ld: TransactionStartTime használata a Get Transaction Data résznél).

A következő adatokat fogjuk kimutatni, és ezekhez mind tartozik egy-egy pivot tábla, amihez a vizualizációt rendeljük:

- **Átlagos robot ciklusidő másodpercben:** az adatvagyon Cycle Time oszlopa kerül a pivot tábla értékek részéhez, átlag értékmező beállítással
- **Feldolgozott tételek száma:** az adatvagyon Cycle Time oszlopa kerül a pivot tábla értékek részéhez, darab értékmező beállítással
- **Összes futási idő órában:** a Cycle Time alapján létrehozunk egy számított mezőt a pivot táblába ezzel a képlettel: $\text{'Cycle Time'}/3600$
- **Hisztogram státusz felosztásban:** az adatvagyon Datum oszlopa kerül a pivot tábla sorok részébe, a Status kerül az oszlopokhoz, valamint szintén a Datum kerül (de bármelyik másik oszlop is kerülhetne) a pivot értékek részéhez, darab értékmező beállítással. A kész pivot táblára pedig egy halmozott oszlop diagramot helyezve megkapjuk a szükséges grafikont.
- **Kördiagram státusz felosztásban:** a pivot tábla sorok részéhez kerül a status, az értékekhez pl a Datum. Az értékek megjelenítéséhez a végösszeg százalékát használtam, majd ehhez a pivothoz hoztam létre a kördiagramot.
- **Kördiagram Business Rule Exception státuszú tételek ok felosztásban:** hasonlóan az előző kördiagramhoz, csak a reason került a pivot sorok területébe, és a szűrőhöz rögzítettem a statust, leszűrve a Business Rule Exceptionre. Majd szintén létrehoztam a kördiagramot.



26. ábra RPA vizualizáció (Vizualizáció_log.xlsx melléklet – saját közlés) Jelmagyarázat: Y-tengely mértékegysége = tranzakció darabszám)

A könnyebb navigálás érdekében elhelyeztem egy szeletelőt és egy idősort a vizualizáció felületére. A szeletelő a Process alapján teszi lehetővé az adatok könnyebb vizualizálását, a idősor pedig lehetőségek az a felhasználó számára a kívánt időszak kiválasztására. Mindkettő esetében beállítottam a kimutatáskapcsolatokat, hogy az összes pivot táblára legyenek hatással.

A visszamérést tekintve szükségünk lesz még az emberek által feladatok ciklusidejére, és ebből már meghatározhatjuk azt az időmennyiséget, amit a robot felszabadít. Itt több metrikát lehet felhasználni ennek kimutatására, de ez nem képezi részét a fejlesztésnek. Ajánlásom ugyanakkor a következő:

$\text{Emberi ciklusidő} * \text{sikeres tételek száma} + \text{emberi ciklusidő} * \text{üzleti kivételek száma} * 0.5.$

A robot ciklusidejét gyakran hasonlítják az emberi ciklusidővel, és próbálnak, véleményen szerint hibásan, a robot ciklusidejével számolni megtakarítás számításnál.

3.12. Információbiztonsági aspektusok

Az információbiztonság itt külön alfejezetben szerepel, viszont a teljes fejlesztési ciklus alatt kiemelt szerepe van. A szoftverrobotokat elsődlegesen és jóhiszeműen pajzsnak tekintjük, hiszen egyik fő oka a felhasználásának és implementálásának, hogy megvédje a vállalatokat a felesleges költségektől. Ugyanakkor nem sétálhatunk el amellett, hogy a robotra úgy tekintünk, mint fegyver. Bármilyen adatbevitelt támogató robot manipulációja esetén a robot a támadás eszköze válik. Ez történhet még az input előállításának oldalon, amikor maga az adat, amit a robot megkap már korrupt, a robot pedig rögzíti ezt (például túlfizetések visszautalása esetén korruptálja a támadó a visszautalandó számlaszámot), de történhet közvetlenül a robot forráskódjában (például ugyan ezt a példát használva, beleégetik a kódba a visszaélésre használt számlaszámot, akár valamilyen feltétel mellett, hogy nehezebben lehessen detektálni).

A robot kódjában is képesek vagyunk egy bizonyos szintig kezelni az ilyen támadásokat (például egy számlaszámot nem használhatunk fel többször és egyedinek kell lennie), de ha a robot forráskódjában történik a támadás, az ellenőrzést az érintett és felhasznált IT rendszer más vetületében kell ellenőrizni. Ezt a meglévő rendszerek működésében vizsgálандó és detektálható, egyfajta csalás felismerés (fraud-detection) módszerrel.

Ilye támadás esetében a lehetséges megoldások egyike a robot hozzáféréseinek deaktiválása, de ha a támadók több felhasználó belépési adatait is megszerezték, és azokat használják fel a robot kódban, a robot géppark hálózatról való teljes leválasztása, szó szerint a tápkábel dugóból való kihúzása jelentheti a megoldást.

Ennek a problémának a teljes mértékű megoldása nem képezi részét a szakdolgozatnak.

4. Vita

A fejlesztés során többször találkoztam azzal a problémával, hogy a lehetséges hibakezelés nem elegendő, és további lehetőségeket illene keresni, hol hibázhat a felhasználó, egyáltalán elhíhetünk-e bármit a felhasználónak, vagy minden általa megadott bemeneti értéket meg kell-e vizsgálnunk, és nem megfelelő, adott esetben tiltott adat feldolgozását a robotnak meg kell tagadnia.

Szintén felmerült, hogy szükséges-e a dispatcher robot számára, hogy az input állományt fájlba mentse az Init state-ben, majd ezt a fájlt olvassa be a Get Transaction Data state-ben, ahelyett, hogy csak a datatable változót közlekedtetné a state-ek között.

Valamint, ha adott az egységes informatikai infrastruktúra és virtuális környezet, valóban szükség van-e a DWH connector esetén mind az OLEDB, mint az ODBC driver használatához.

5. Következtetések

Pont ennek a fejlesztésnek az apropóján alakítottam ki a DWH connectort, és oldottam meg, hogy a robot képes legyen a vállalati adattárházból a háttérben adatot írni és olvasni. A korábbi működés GUI alapon dolgozott lekérdezésekkel, amik eredménye egy lementett Excel fájl volt, amit szükséges volt beolvasni. Az új komponens révén erre nincs szükség, és valóban felesleges az adatot fájlba írni, majd beolvasni, itt elég lenne az adattáblát használni. Ugyanakkor bizonyos folyamatok esetében az üzleti oldal elvárja, hogy számukra is elérhető legyen, a robot mit, mikor olvasott be feldolgozásra, így az adatok kimentése más folyamatok esetében szükséges lehet.

A hibakezelést tekintve azt a megállapítást hoztam, hogy a mivel a folyamatban az üzemeltető felhasználók száma alacsony (2 személy), akik kellő tudással és előképzettséggel rendelkeznek, a kimutatások hónapról-hónapra, hétről-hétre azonosak, valamint a tesztelés során tapasztalt üzleti hibákból tanultakat alkalmazva a robot működése kellően biztonságos. Rosszhiszemű felhasználást kizárhatjuk, hiszen, ha adatlopás lenne a cél, akkor azt megtehetnék közvetlenül a saját maguk által előállított adatokkal is. Ugyanakkor a cégen kívüli domain címekre küldendő kimutatást illendő és szükség szerű hibaágra tenni (akár elgépelés miatt is).

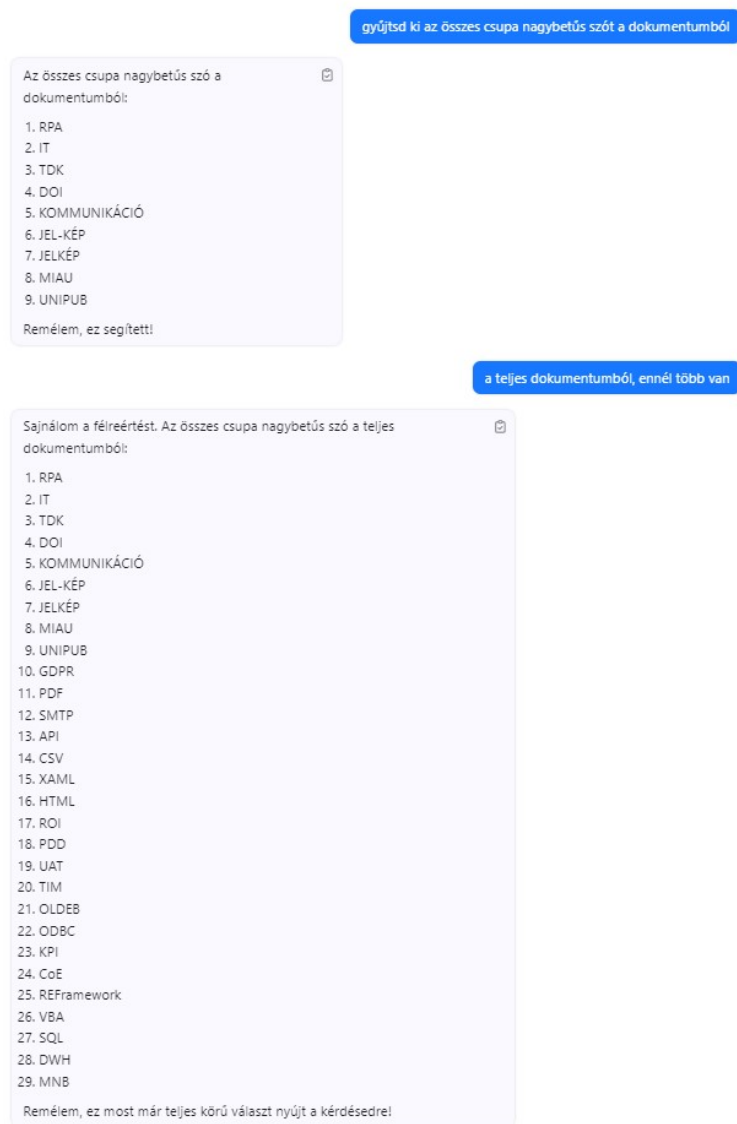
A 2 database driver párhuzamos használata ugyan jelen esetben felesleges (minden virtuális gép OLEDB-t használ), ha bármikor a jövőben az OLEDB megoldás már nem támogatott eljárás, akkor kódváltás nélkül képes a robot tovább működni, vagy hiba esetén új virtuális környezet kialakításánál, ha ODBC driver kerül csak telepítésre, akkor is képes a robot a feladat végrehajtására.

6. Jövőkép

A robot élesítése után a megoldást a vállalat további területei számára is bemutatjuk, és igény esetén a keretrendszert más terület számára is kiajánljuk üzletvitel szerű működésre. A többi terület esetleges igényeit pedig a meglévő keretrendszerbe implementálnánk, majd ugyan ezt a funkciót az eredti terület számára is elérhetővé tesszük. Ezáltal a robot megtérülése is lépcsőzetesen gyorsulna, hiszen egy-egy új terület számára, már csak adaptálni kell a meglévő működést.

Így a robot akár hosszútávon is képes ellátni különböző területek adatszolgáltatási feladatát, de adott esetben, ha új reporting szoftver kerül bevezetésre, akkor az átállás is könnyebb lehet, hiszen a robot révén egységesítettük és standardizáltuk az adatszolgáltatás folyamatát, valamint részlegesen dokumentációt is be tudunk mutatni és erre nem kell külön erőforrást áldozni, hogy a területek maguk, egymástól elzártan silókban készítsék el ugyan ezt.

A fejlesztés jövőképét tekintve, a mesterséges intelligencia térnyerésével, és a vállalatok adaptálódási kedvének növelésével képes lehet teret nyerni. A szakdolgozat tekintetében például a rövidítések jegyzékének összeállításához is a ChatGPT-t, illetve ChatPDF-et hívtam segítségül, hogy a dokumentumból gyűjtsék ki az összes csupa nagybetűvel írt szavakat, de tény, hogy csak másodjára sikerült, és az URL-ekben található szövegrészeket is kiszedte, de az utasítást pontosítva (például szóköz előzze meg a nagybetűs szavakat) ez tovább javítható. Sőt, akár egy önálló programot is készíthetnénk, ami a szakdolgozatírást ilyen fajta keretek között támogatja.



27. ábra Mesterséges intelligencia felhasználása a szakdolgozatban kontext-free módon

7. Összefoglalás

A riporting keretrendszer kiépítése projektet összességében sikeresnek tekintem. A projekt keretében az alábbiak valósultak meg maradéktalanul:

- Excel (xlsx) és csv formátumú fájlok előállítása
- Titkosítás, lapfülvédelem- kezelése xlsx típusú fájlknál
- Kimutatás formázás xlsx típusú fájlknál
- SMTP szerveren való email küldés
- Hálózati meghajtóra történő adat export

Ugyanakkor a további lehetőségeket a jövőben vizsgálni fogom, és szabad erőforrás esetén javaslom azok implementálását:

- További lehetséges hibaokok kiszűrése
- További adatkimentések vizsgálata, például más adatbázisba (helyi felhasználók által üzemeltetett Access adatbázis) való beillesztés
- Email jelzés küldése akkor is, ha nem kell csatolni az adatokat (de jelezni az üzlet számára, hogy a kért kimutatások előálltak)
- További kimutatás paraméterek implementálása (bővebb, felhasználóbarátabb formázás, vállalati logó elhelyezése, összefoglaló lapfűl készítése)

Rövidítések jegyzéke

- RPA: Robotic Process Automation, robotikus folyamatautomatizáció
- TDK: Tudományos Diákkör
- CoE: Center of Excellence, kompetencia központ
- REFramework: Robotic Enterprise Framework
- VBA: Visual Basic for Applications
- SQL: Structured Query Language
- DWH: data warehouse, adattárház
- SMTP: Simple Mail Transfer Protocol
- API: Application Programming Interface
- CSV: Comma-Separated Values, vesszővel tagolt értékek
- XAML: Extensible Application Markup Language
- HTML: HyperText Markup Language
- GDPR: General Data Protection Regulation
- PDF: Portable Document Format
- KKV: kis- és középvállalkozások
- MNB: Magyar Nemzeti Bank
- ROI: return on investment, befektetésarányos megtérülés
- PDD: process definition document
- UAT: user acceptance test
- TIM: Tivoli Identity Manager (IBM)
- OLDEB: Object Linking and Embedding, Database
- ODBC: Open Database Connectivity
- KPI: key performance indicator, kulcsfontosságú teljesítménymutató.

Ábrajegyzék

1. ábra Robot, mint szakértői rendszer (Forrás: Kiértékelési_fa_szakértői_rendszer.xlsx és Szakértői_rendszer.xaml mellékletek - saját közlés) 17
2. ábra Wikipedia história összehasonlítás robottal és JavaScript-tel..... 20
3. ábra Gantt chart (Forrás: Gantt-RPA.xlsx melléklet - saját közlés)..... 25

4. ábra ChatGPT által készített xaml állomány (Forrás: https://chat.openai.com/c/88e3ba2e-6af2-41f1-b049-91f41b14e9f5 , Folyamatszervezés riport keretrendszer\Performer_DB.Activities\ ChatGPT_xml.xaml melléklet)	26
5. ábra ChatGPT által készített VBA makró (Forrás: https://chat.openai.com/c/51b3f616-cc2b-49fa-9814-259d852059af , ChatGPT_VBA.txt melléklet)	27
6. ábra Input adattábla szerkezete és mezők magyarázatai (Forrás: RPA_Excel_Kombinatorikai_tér.xlsx melléklet, RAW lapfűl – saját közlés)	30
7. ábra SMTP email küldés csatolmány nélkül (Forrás: Library_Emailing\SMTP\ Send_Email.xaml melléklet – saját közlés).....	38
8. ábra Bemeneti értékek és típusaik (Forrás: Library_Emailing\SMTP\ Send_Email.xaml melléklet – saját közlés)	39
9. ábra ChatGPT válasza a UiPath és Oracle adatbázis összekapcsolásához (Forrás: https://chat.openai.com/c/11bb95d5-bb56-46bd-8ce7-51b242aa97a5)	40
10. ábra DWH kapcsolat kiépítése (Forrás: Library_DWH\DWH_Connection\Estabilish_DWH_connection.xaml melléklet – saját közlés)	41
11. ábra Config Excel I. (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Data\Config.xlsx – saját közlés)	43
12. ábra Config Excel II. (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Data\Config.xlsx – saját közlés)	43
13. ábra REFramework state-machine ábrája ((Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Main.xaml – saját közlés))	45
14. ábra Framework\KillAllProcesses.xaml (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Framework\KillAllProcesses.xaml melléklet – saját közlés)	46
15. ábra Framework\InitAllSettings.xaml (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Framework\ InitAllSettings.xaml melléklet – saját közlés)	47

16. ábra Framework\InitAllApplications.xaml (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Framework\ InitAllApplications.xaml melléklet – saját közlés)	48
17. ábra Hozzáférési adatok menedzselése az Orchestrator-ban (Forrás: saját közlés)	49
18. ábra TableDownload.xaml bemenő paraméterei (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\TableDownload.xaml melléklet – saját közlés)	52
19. ábra TableDownload.xaml működése (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\TableDownload.xaml melléklet – saját közlés)	53
20. ábra SMTP küldés paraméterei (Forrás: Library_Emailing\SMTP\Send_Email_SingleAttachment.xaml melléklet – saját közlés)	56
21. ábra Garbage Collector Collect és WaitForPendingFinalizers metódusok (Forrás: Folyamatszervezés riport keretrendszer\Performer_DB.Activities\Process.xaml melléklet – saját közlés)	57
22. ábra Queue létrehozása Orchestrator-ban (Forrás: saját közlés)	58
23. ábra Publish a Studio-ból (Forrás: saját közlés)	60
24. ábra Ütemezés az Orchestrator-ban (Forrás: saját közlés)	62
25. ábra Az adatvizualizáció adatvagyona (Forrás: Vizualizáció_log.xlsx melléklet – saját közlés)	63
26. ábra RPA vizualizáció (Vizualizáció_log.xlsx melléklet – saját közlés) Jelmagyarázat: Y- tengely mértékegysége = tranzakció darabszám)	64
27. ábra Mesterséges intelligencia felhasználása a szakdolgozatban kontext-free módon	70

Hivatkozásjegyzék

- What is robotic process automation? <https://www.uipath.com/rpa/robotic-process-automation>, 2024.03.26
- 4 percben: amit az RPA-ról tudni érdemes <https://www.controllingportal.hu/amit-az-rpa-rol-tudni-erdemes/>, Bitó János, 2018.03.20.
- Digitalizációs megoldások az ügyvitelben Musinszki Zoltán, A digitalizáció hatása a back office munkára, In: Veresné, Somosi Mariann; Sikos, T. Tamás (szerk.) A fenntarthatóság holisztikus megközelítésben, Budapest, Magyarország: Akadémiai Kiadó (2023) 501 p. pp. 107-122., 16 p.
- RPA felhasználási területei https://hu.wikipedia.org/wiki/Robotiz%C3%A1lt_folyamatautomatiz%C3%A1l%C3%A1s, Wikipedia, 2024.03.26
- Demokratizált fejlesztés <https://www.linkedin.com/pulse/rpa-lead-us-automation-duality-stefan-schnell>, Stefan Schnell, LinkedIn, 2020.11.30
- Szoftverrobotok a bankokban. Terjed a robotizáció (RPA) a pénzügyi szektorban is <https://fintechzone.hu/szoftverrobotok-rpa-segitik-penzugyeink-intezeset-is/>, Urbán Anikó, 2023.10.09
- Intelligens dokumentumfeldolgozás <https://goodroid.hu/intelligens-dokumentumfeldolgozas/>, 2024.03.26
- Chatbotok és az RPA <https://www.automationanywhere.com/company/blog/rpa-thought-leadership/why-combine-chatbots-and-rpa>, 2024.03.26
- 4. ipari forradalom, és a robotok hatása a munkaerőpiacra <https://mnbintezet.hu/tenyleg-elveszik-a-gepek-az-emberek-munkajat/>, Kotymán Alex, 2023.03.20
- Why writing a Process Definition Document for RPA is not as easy as most people think <https://www.linkedin.com/pulse/why-writing-process-definition-document-rpa-easy-most-papp/>, Papp Bálint László, LinkedIn, 2021.03.28
- Why automate a bad process? <https://isg-one.com/articles/why-automate-a-bad-process>, 2024.03.26

- Hogyan kerülnek a robotok az irodába? <https://lean.org.hu/eszkozok/hogyan-kerulnek-robotok-az-irodaba/>, Nagy Richárd, 2021
- Szakértői rendszer <https://miau.myx.hu/miau2009/index.php3?x=e0&string=szak%C3%A9rt%C5%91i%20rendszer>, 2024.03.29
- Üzleti intelligencia? Robot intelligencia? Mesterséges intelligencia? Boros Tamás - Filippov Gábor (szerk.): Magyarország 2030. Jövőkép a magyaroknak (Budapest, 2020)
- RPA és az információbiztonság ChatGPT, <https://chat.openai.com/c/4e163902-4986-448b-b174-b5d62177f230>, 2024.03.28
- ChatPDF felhasználásának ötlete a szakdolgozatban Molnár Csanád Iván: Online Dietetikus Szakértő Mesterséges Intelligencia segítségével https://miau.myx.hu/miau/307/odessai_mcs.pdf
- Korábbi egyetemi dokumentációm, beleértve a TDK publikációm: <https://miau.myx.hu/miau2009/index.php3?x=e0&string=nagy%20rich%C3%A1rd>, 2024.04.04
- Tudományos írás (academic writing skills): https://unipub.lib.uniconvinius.hu/4649/1/JelKep_2018_4_Deli_Eszter_Retvari_Marton.pdf, 2018.04

Tartalomjegyzék

Robotikus folyamatautomatizáció implementálása nagyvállalati környezetben	2
Kivonat	3
Kulcsszavak.....	3
Title of document	3
Abstract	3
Keywords	4
1. Bevezetés.....	5
1.1. Köszönetnyilvánítás.....	5
1.2. Célok.....	5
1.3. Feladatok	5
1.4. Motivációk.....	6
1.5. Célcsoportok.....	7
1.6. Hasznosság	7
1.7. A dolgozat szerkezetéről	8
2. Szakirodalmi háttér	12
2.1. Gazdasági, vállalatirányítási szempontból	12
2.2. IT szempontból.....	13
2.3. HR szempontból	15
2.4. Üzleti szempontból.....	15
2.5. A mesterséges intelligencia szempontjából.....	17
2.6. A probléma/jelenség története	19
2.7. A szakdolgozati téma és a szak tantárgyainak kapcsolata.....	20
3. Saját fejlesztések	25
3.1. Követelményspecifikáció	27
3.2 Megvalósítási tanulmány	30

3.2.1. Dispatcher	31
3.2.1.1. Init	31
3.2.1.2. Get Transaction Data.....	32
3.2.1.3. Process Transaction.....	32
3.2.1.4. End	33
3.2.2. Performer	33
3.2.2.1. Init	33
3.2.2.2. GetTransactionData.....	34
3.2.2.3. Process Transaction.....	34
3.2.2.4. End	36
3.3 Felhasználó, valamint jogosultság igénylés.....	36
3.4. RPA könyvtárak elkészítése	37
3.4.1. SMTP Connector Könyvtár	37
3.4.2. DWH Connector Könyvtár	39
3.5. Robot kód fejlesztése.....	42
3.6. Queue-k létrehozása	57
3.7. Asset-ek létrehozása	58
3.8. Robot kód üzleti tesztelése	58
3.9. Robot élesítése.....	60
3.10. Robot üzemeltetése.....	62
3.11. Adatvizualizáció, riportok, visszamérés.....	63
3.12. Információbiztonsági aspektusok	65
4. Vita	67
5. Következtetések	68
6. Jövőkép.....	69
7. Összefoglalás.....	71

Rövidítések jegyzéke.....	72
Ábrajegyzék	72
Hivatkozásjegyzék	75
Tartalomjegyzék.....	77