

**M A S A R Y K O V A
U N I V E R Z I T A**

FAKULTA INFORMATIKY

**Vytvoření modulu v ERP Business
Central**

Diplomová práce

KAROLÍNA HEČOVÁ

Brno, jaro 2023

**M A S A R Y K O V A
U N I V E R Z I T A**

FAKULTA INFORMATIKY

Vytvoření modulu v ERP Business Central

Diplomová práce

KAROLÍNA HEČOVÁ

Vedoucí práce: Ing. Leonard Walletzký, Ph.D.

Katedra počítačových systémů a komunikací

Brno, jaro 2023



Prohlášení

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracovala samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používala nebo z nich čerpala, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Karolína Hečová

Vedoucí práce: Ing. Leonard Walletzký, Ph.D.

Poděkování

Zde bych ráda poděkovala sama sobě, že jsem to letos nenechala na poslední týden před odevzdáním. Dále bych chtěla poděkovat panu Ing. Leonardu Walletzkému, Ph.D., že se mě ještě jednou ujal i přesto, že už se této oblasti oblasti nechce moc věnovat. Ale hlavně bych chtěla poděkovat kolegovi, který do mě ryl tak dlouho, až jsem k tomu nakonec sedla ...

Shrnutí

Tato práce se zabývá vývojem samostatného modulu sloužícího k integracím souborů pro Enterprise Resource Planning (ERP) systém Business Central (BC) od firmy Microsoft. Teoretická část se věnuje stručné teorii ERP, následně se můžeme dozvědět něco přímo o BC. V praktické části projdeme procesem vývoje nového modulu. Zanalyzujeme problém výměny souboru mezi cloudovým prostředím a lokálním úložištěm. Následuje několik digramu k samotnému návrhu cílového řešení. Poté přichází implementační fáze společně s testovací. V závěrečné části si ukážeme několik příkladů z živého provozu, následuje zpětná vazba od zákazníka.

Klíčová slova

ERP systém, modul, Business Central, on-premise, cloud, zpracování souborů, ...

Obsah

Úvod	1
1 ERP systémy	2
1.1 Definice	2
1.1.1 Oblasti ERP systému	2
1.2 Výhody a nevýhody	2
1.2.1 Výhody	3
1.2.2 Nevýhody	3
1.3 Customizace	4
1.3.1 Výhody a nevýhody	4
1.3.2 Nejčastější důvody ke customizaci	4
2 Microsoft Dynamics 365 - Business Central	6
2.1 Podrobněji o BC	6
2.2 Výhody BC	7
2.3 Popis architektury	8
2.3.1 Typy objektů BC	9
3 Analýza – Problém výměny souborů	11
3.1 Standardní možnosti řešení	11
3.2 Specifika oblasti	15
3.3 Cílové řešení	16
4 Návrh	17
4.1 Funkční návrh	17
4.1.1 Use Case specifikace	18
4.2 Procesní návrh	23
4.2.1 Import souborů	23
4.2.2 Export souborů	26
4.3 Datový návrh	28
5 Implementace	30
6 Testování	43
6.1 Problémy objevené při testování	44
6.1.1 Zpracování všech souborů v adresáři	44

6.1.2	Duplicitní importované soubory	44
6.1.3	Ošetření ručního importu souboru	45
7	Provoz	46
7.1	Nastavení komunikací	46
7.2	Položka protokolu integrace	47
7.3	Automatické úlohy	47
8	Vyhodnocení	49
8.1	Zpětná vazba	49
	Závěr	53
	Bibliografie	54
A	Přílohy	55

Úvod

Klíčovou roli většiny firem hrají podnikové informační systémy. Možná je znáte spíše pod jejich anglickým názvem ERP. Pomáhají firmám optimalizovat procesy a tím i šetřit peníze a drahocenný čas.

Práce se zaměřuje na ERP systém od firmy *Microsoft* a to konkrétně na jejich *BC*, který je součástí produktové řady *Microsoft Dynamics 365*. Ten již v dnešní době obsahuje rozšíření specializované pouze pro Česko, bohužel většina firem i tak potřebuje menší či větší modifikace tohoto systému, aby mohli efektivněji integrovat dílčí procesy. A právě postup na vypracování jednoho takového rozšíření se pokusím trochu detailněji přiblížit.

V první části si stručně projdeme ERP systémy, jejich klíčové vlastnosti, proč jsou tak žádané. Poté již přejdeme k samotnému *BC* a opět si stručně popíšeme specifika právě tohoto systému.

Po úvodní teorii přejdeme k jádru samotné práce. Dojde k nastíní dílčího problému, který se pokusíme vyřešit napřed standardní cestou, která ale z různých důvodů nemusím splňovat konkrétní požadavky zákazníku. Proto následně přejdeme k samotnému procesu sběru požadavků od zákazníků, které vyústí k definování cílů výsledného rozšíření.

Následně projdeme fázemi projektu jako je *návrh*, samotná *realizace*, *testování*, *provoz* u zákazníka - z počátku v testovacím prostředí, poté dochází k přesunu na živé prostředí. Závěrem se pokusím zhodnotit výsledky práce, její přínosy. Přidám i zpětnou vazbu přímo od zákazníka, ze které možná vyplynou body na budoucí vylepšení tohoto modulu.

Práce byla vytvořena ve spolupráci s firmou *Essence International s.r.o.*, kde v současné době pracuji převážně jako vývojář.

1 ERP systémy

V krátkosti k definici ERP systémů obecně. Jelikož se jedná o komplexní systém, tak těchto definic existuje, troufám si říct, až nekonečné množství. Jedno ale zmiňuje většina z těchto definic a to, že se jedná o systém integrující procesy, které jsou klíčové pro běh firmy. Zároveň se ERP snaží tyto procesy zefektivňovat.

1.1 Definice

ERP je systém, software, který integruje interní a externí správu informací napříč celou společností. Účelem ERP je usnadnit tok informací mezi všemi podnikovými funkcemi uvnitř daných organizací. Systém ERP je určen k obsluze všech činností organizace v rámci jediného softwarového systému. Výstupy jedné oblasti tak mohou být použity jako vstupy jiné oblasti, aniž by bylo nutné data ručně duplikovat. (1)

1.1.1 Oblasti ERP systému

Systém by šlo rozdělit na 4 hlavní oblasti, kde se každá oblast skládá z řady různých modulů s funkcemi pro danou oblast: (2)

- *finance* – závazky a pohledávky, řízení hotovosti a předpovědi, účetnictví, dlouhodobý majetek, analýza ziskovosti, ...
- *personalistika* – evidence odpracované doby, mzdy, plánování pracovníků, cestovní výlohy, ...
- *výroba a logistika* – sklady a řízení výroby, plánování výroby, nákup a příjem zboží, řízení projektů, hodnocení dodavatelů, ...
- *marketing a prodej* – zpracování zakázek, ceny, řízení prodeje, plánování prodeje, servis, ...

1.2 Výhody a nevýhody

Nyní se podíváme na seznam, který si většina firem asi dá dohromady, než si tento systém pořídí. Ne pro všechny je množství výhod rozhodujícím faktorem a často se musí zamyslet, zda je to výhodné zrovna

pro ně a zda nevýhody nepřeváží. Někdy je totiž návratnost těchto systémů až v dlouhodobém horizontu. (3)

1.2.1 Výhody

- zefektivnění a zrychlení procesů (ekonomických)
- centralizace dat – dostupnost přesných a konzistentních dat, sdílení dat
- snížení chybovosti
- zvýšení bezpečnosti
- úspora investic do IT (v dlouhodobém horizontu)
- rychlejší výstupy pro vedení firmy
- podpora účetnictví
- zvýšení konkurenceschopnosti
- zrychlení schvalování (např. plateb)
- možnost propojení s dodavateli a odběrateli

1.2.2 Nevýhody

- vysoká cena – licenční poplatky
- další náklady – údržba, školený, rozšiřování
- závislost na dodavateli
- špatná ovladatelnost systému
- funkčnost neodpovídá potřebám

1.3 Customizace

Přestože z dlouhodobého hlediska nabízí ERP systému spoustu benefitů, většinou je potřeba částečná customizace tzv. krabicového řešení pro potřeby dané firmy. Customizace obnáší zásah do kódu, jsou náročné jak časově tak i finančně a mohou přinášet prostor pro potenciální chyby v softwaru.

Studie společnosti Panorama Consulting ukázala, že 90 % systémů ERP má provedeny alespoň drobné úpravy, zatímco 36 % má úpravy, které zahrnují úpravu více než 50 % kódu. Pouze 2 % jsou kompletně přizpůsobena nebo vytvořena na zakázku. (4)

1.3.1 Výhody a nevýhody

S přibývajícimi změnami se stává ERP systém čím dál víc komplexnější. Je také finančně náročnější na údržbu, případný upgrade už nemusí být v určité fázi ani úspěšně proveditelný. Právě nemožnost další úpravy se stává jedním z důvodů na případnou změnu ERP systému za nový. (4)

Světlá stránka procesu customizace je, že po úspěšném provedení budete mít systém navržený na míru pro potřeby vašeho podnikání. Customizace dávají podnikům určitou konkurenční výhodu právě možností upravit nebo přidat nové části do ERP. Přestože většina chce implementaci ERP systému zvládnout pouze pomocí základního nastavení, správnou konfigurací a případnou personalizací, většina nakonec nějakou zásadnější změnu v systému provede. (4)

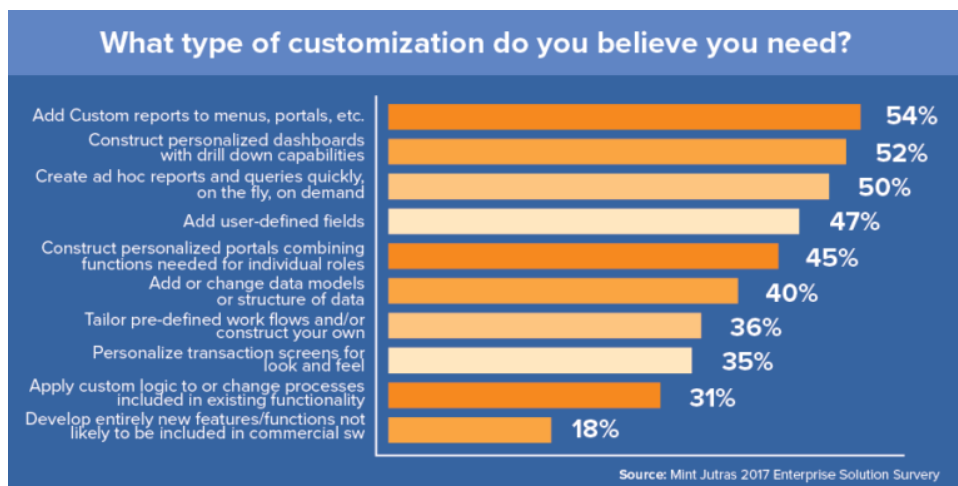
1.3.2 Nejčastější důvody ke customizaci

V článku od společnosti Mint Jutras proběhl průzkum na téma: „Jaký typ customizace podle vás potřebujete?“. Mezi odpověďmi se objevilo například: (5)

- vlastní reporty do menu, portálů, atd.
- ad hoc sestavy pro rychlé dotazy za běhu a na vyžádání
- uživatelsky definovaná pole
- personalizované portály kombinující funkce pro jednotlivé role

- řídicí panel na míru s možností rozbalení

Na posledním místě se pak umístil požadavek na vývoj kompletně nové funkcionality, která v systému chyběla.



Obrázek 1.1: Průzkum typů customizace, Zdroj: Carter Lloyds, 2017 (5)

2 Microsoft Dynamics 365 - Business Central

Jak již název napovídá, tak BC je součástí produktové řady *Microsoft Dynamics 365*, ta ve svém portfoliu zahrnuje software pro řízení vztahů se zákazníky – Customer Relationship Management (CRM) a ERP software. BC je cloudová aplikace pro plánování podnikových zdrojů (ERP), která je poskytována prostřednictvím modelu *Software as a Service*¹ (SaaS) prostřednictvím certifikovaných partnerů *Cloud Solution Provider* (CSP). (6)

2.1 Podrobněji o BC

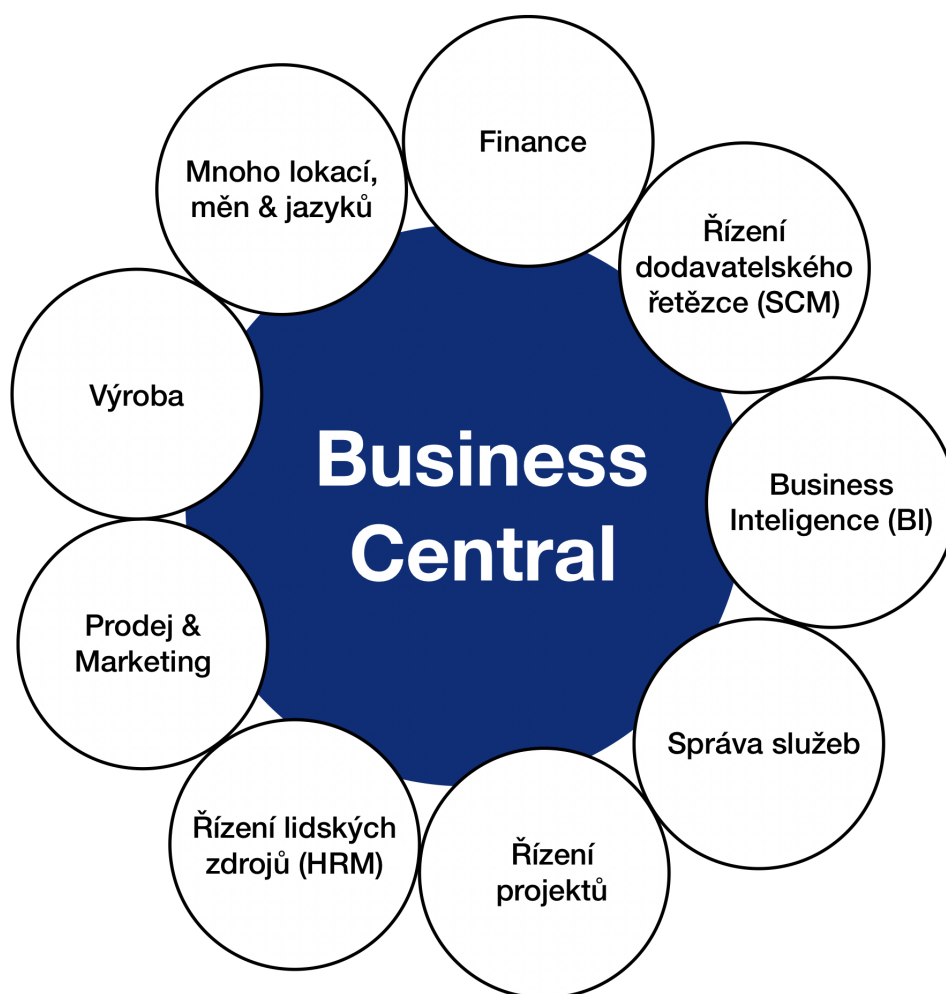
Pro BC platí principy obecného ERP systému popsané výše, integruje interní a externí data v různých funkčních oblastech. Jednotlivé oblasti jsou v grafické podobě znázorněny níže 3.1. Díky dobré integraci jednotlivých funkčních oblastí poskytuje sadu nástrojů pro zpracování informací, které lze zadat jednou a následně je použít mnoha způsoby.

Aplikace BC má uživatelské rozhraní (UI) přizpůsobené dané roli. V BC existují čtyři instance klienta na míru daným rolím – *Role Tailored Client* (RTC). Konkrétně pro Windows, webovou interakci, použití na tabletu a jako klienta pro mobilní zařízení. Obecně se na jednotlivé klienty odkazuje většinou podle toho, kde a jak se zobrazují (Windows, web, tablet atd.). Klient systému Windows je k dispozici pouze pro on-premise² instalace.

BC je modulární ERP systém a vývoj nových funkcí zde probíhá pomocí tzv. *extensions*. Jedná se o rozšíření, které je nezávislé může být nezávislé na základní aplikaci od firmy *Microsoft* a zároveň je samostatně zkompileovatelné a nasaditelné. Programování těchto rozšíření probíhá pomocí programovacího jazyka *AL Code*, jenž by se dal přirovnat ke směsi známějších jazyků jako je Pascal, C# a JavaScript.

1. Software as a Service – software pronajatý jako služba

2. on-premise – lokálně nainstalovaný software



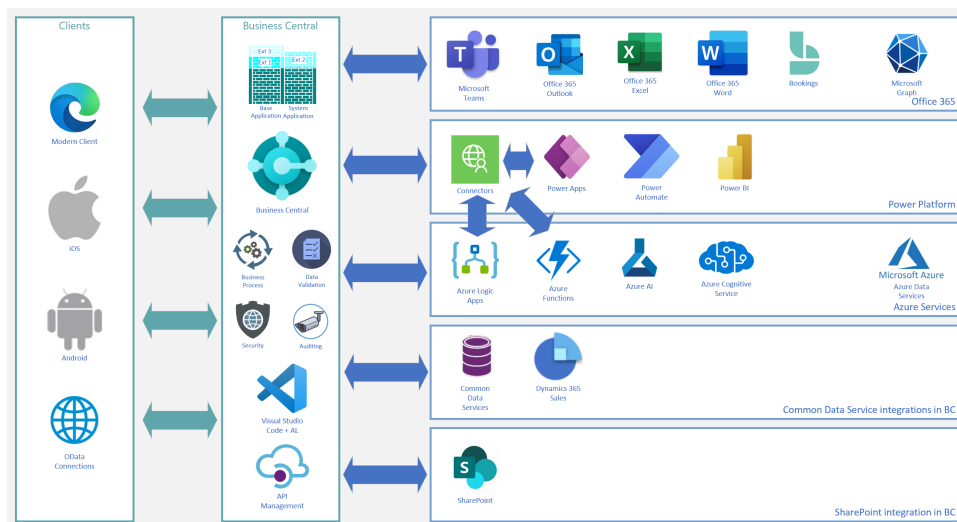
Obrázek 2.1: Funkční oblasti BC, Zdroj: vlastní zpracování

2.2 Výhody BC

V této části si shrneme několik důvodů, proč se firma, ve které v současné době pracuji, zabývá implementací právě tohoto ERP systému. Jsou to zároveň důvody, díky kterým se k tomuto systému přiklání i zákazníci a právě proto se vyplatí implementovat obecný modul, kterým se zabývám níže v této práci.

Důvody k výběru *Dynamics 365 Business Central* (7):

- prověřený systém stovkami firem, který vychází z oblíbeného předchůdce ERP systému Navision (s modernizovaným designem)
- shromažďuje data s každého koutu firmy
- uživatelské prostředí se podobá ostatním softwarům od firmy *Microsoft*
- je plně integrovaný s dalšími aplikacemi *Microsoft*, od *MS Office* po CRM systém *Customer Engagement*



Obrázek 2.2: Integrace BC, Zdroj: Aleksandar Totovic, 2020 (8)

2.3 Popis architektury

Z pohledu vývojáře se BC skládá z téměř šesti tisíc potencionálně přizpůsobitelných objektů. Tyto objekty je možné upravovat pomocí *Visual Studio Code* (VS Code), zde můžeme také vytvářet objekty nové. (9)

2.3.1 Typy objektů BC

- **Table** – slouží k definici datové struktury a k ukládání záznamů
- **Page** – slouží k formátování a zobrazování dat na základě jednotlivých typů klientů a uživatelských rolí
- **Reports** – slouží k zobrazení dat uživateli v tištěné podobě, a to buď na obrazovce (režim náhledu), nebo prostřednictvím tiskového zařízení, objekty sestav mohou také aktualizovat data v procesech se zobrazením dat nebo bez něj
- **Codeunits (CU)** – slouží k uchování kódu, který může být využit jinými objekty, jsou strukturovány do segmentů kódu nazývaných *procedury* (ekvivalent definice funkcí)
- **Query** – slouží k extrakci dat z jedné nebo více tabulek, umožňují provádět výpočty a vypisovat je ve formě nové datové struktury (do grafů, Excelu, XML, OData, ...), lze je použít jako nepřímé zdroje pro stránky (page) a sestavy (report)
- **Enum** – slouží k definici konstant, mohou být použity jako zdroj hodnot pole v tabulce nebo jako proměnná
- **XMLport** – slouží k importu a exportu dat do/z externích souborů, struktura externích souborů může být ve formátu XML nebo jiných
- **Profile** – slouží k definici centra rolí (domovská stránka) a k přizpůsobení stránek pomocí skupin uživatelů
- **Control add-in** – slouží k vizualizaci pomocí HTML5 založené na datech v BC, mohou odesílat události, které lze použít ke spuštění dalšího kódu
- **Table extension** – slouží k vytváření doprovodných tabulek, které jsou propojeny s tabulkami definovanými společností *Microsoft* nebo jinou společností
- **Page extension** – slouží k přidávání nebo skrývání prvků na existujících stránkách

- **Page customization** – slouží k úpravě a přesouvání prvků na stávajících stránkách, jedná se převážně o nástroj pro použití koncovými uživateli
- **Report extension** – slouží k přidávání nebo skrývání prvků na existujících reportech, k definici nových datových položek nebo k definici nového rozložení reportu
- **Enum extension** – slouží k přidávání nových konstant do existujícího výčtů
- **Interface** – definice rozhraní funkcí a jejich parametrů, které lze následně implementovat různými CU

3 Analýza – Problém výměny souborů

S přechodem BC do prostředí cloudu vznikl nový potencionální problém s výměnou souborů. Zatímco *on-premise* instalace má přístup k lokálním zdrojům a souborovému systému, tak *SaaS* verze BC tuto možnost nemá, protože všechny procesy běží v datových centrech firmy *Microsoft*.

3.1 Standardní možnosti řešení

Pro *on-premise* řešení je možné používat proměnné typu `File`. Nad tou mohou být provedeny běžné metody správy souborů, např. `Create`.

Kód 3.1: Příklad funkce se založením souboru

```
1 procedure CreateFile()
2 var
3     MyFile: File;
4 begin
5     MyFile.Create('C:\Folder\File.txt', TextEncoding::MSDos);
6 end;
```

Pokud je vývoj určený i pro cloudové prostředí, tak se ve vývojovém prostředí *VS Code* objeví error s příslušnou hláškou a kód nelze zkompileovat. Určení, zda je modul určený i pro cloudová prostředí lze nastavit v souboru *app.json*, který je součástí každého modulu. Zde jsou informace jako unikátní *id* nového extensionu, jeho název, verze, ... Zde je potřeba specifikovat přidat řádek `"target": "Internal"` a kód uvedený výše by se nám podařilo zkompileovat.

Podobnou hlášku od kompilátoru bychom dostali i v případě, pokud bychom se pokoušeli používat standardní *CU File Management* a její procedury.

Pro cloudové řešení se k práci se soubory používají proměnné typu `InStream` a `OutStream`. Soubory lze ukládat do proměnných typu `Blob`, jako je například pole *Picture* na kartě zboží v následujícím příkladu.

Kód 3.2: Příklad funkce pro nahrání obrázku k danému zboží

```

1 procedure ImportItemPicture(var Item: Record Item)
2 var
3     Instream: InStream;
4     FileName: Text;
5 begin
6     if UploadIntoStream('', '', '.*.*', FileName, Instream) then
7         begin
8             Clear(Item.Picture);
9             Item.Picture.ImportStream(Instream, FileName);
10            Item.Modify();
11        end;
12 end;

```

Pro stažení souboru v cloudovém prostředí je potřeba použít standardní funkci s následujícími parametry, která vrací hodnotu typu boolean:

```

[Ok := ] File.DownloadFromStream(InStream: InStream,
    DialogTitle: String, ToFolder: String, ToFilter: String,
    var ToFile: Text)

```

Kód 3.3: Příklad funkce pro stažení obrázku z daného zboží

```

1 procedure ExportItemPicture(var Item: Record Item)
2 var
3     Instream: InStream;
4     FileName: Text;
5 begin
6     if Item.Picture.HasValue() then begin
7         Item.CalcFields(Picture);
8         Item.Picture.CreateInStream(Instream);
9         FileName := Item.No. + '.jpg';
10        DownloadFromStream(Instream, 'Save file', '', 'All
            Files ( *.* )| *.*', FileName);
11    end;
12 end;

```

Jako další možnost pro výměnu souborů slouží XMLports. Definice XMLportu obsahuje parametr, jakým směrem budou proudit data do BC a to konkrétně *Import*, *Export*, *Both*. Data mohou být ve formátu XML nebo CSV (*text*). XMLport nelze z BC zavolat na přímo, je potřeba ho volat z objektu typu Page nebo CU.

Příklad definice XMLportu naleznete níže. Zde jsou funkce, které mohou být využity pro export / import dat.

Kód 3.4: Příklad funkce pro import dat pomocí XMLportu

```

1 procedure RunXMLportImport()
2 var
3     FileInStream: InStream;
4     FileName: Text;
5 begin
6     UploadIntoStream('','',',',FileName,FileInStream);
7     Xmlport.Import(Xmlport::MyCustomerXmlport, FileInStream);
8     Message('Import done successfully!');
9 end;
```

Kód 3.5: Příklad funkce pro export dat pomocí XMLportu

```

1 procedure RunXMLportExport()
2 var
3     TempBlob: Codeunit "Temp Blob";
4     FileName: Text;
5     FileOutputStream: OutStream;
6     FileInStream: InStream;
7     outputFileName: Text;
8 begin
9     TempBlob.CreateOutStream(FileOutputStream);
10    Xmlport.Export(Xmlport::MyCustomerXmlport, FileOutputStream);
11    TempBlob.CreateInStream(FileInStream);
12    outputFileName := 'MyCustomersXmlport.xml';
13    DownloadFromStream(FileInStream,'','',',',outputFileName);
14    // The output is saved in the default download folder
15 end;
```

Kód 3.6: Příklad XMLportu pro výměnu dat na tabulce Customer

```
1 xmlport 50100 MyCustomerXmlPort
2 {
3     Direction = Both;
4     Format = VariableText;
5     TextEncoding = UTF8;
6     FieldSeparator = ',';
7     RecordSeparator = '<NewLine>';
8
9     schema
10    {
11        textelement(NodeName1)
12        {
13            tableelement(Customer; Customer)
14            {
15                SourceTableView = where(Address = FILTER(<>''));
16
17                textelement(CustomerNoLbl)
18                {
19                    trigger OnBeforePassVariable();
20                    begin
21                        CustomerNoLbl := Customer.FieldCaption("
22                            No.");
23                    end;
24                }
25                fieldattribute(No; Customer."No.")
26                {
27                }
28                fieldattribute(Name; Customer.Name)
29                {
30                }
31                fieldattribute(Address; Customer.Address)
32                {
33                }
34            }
35        }
36    }
```

3. ANALÝZA – PROBLÉM VÝMĚNY SOUBORŮ

Výměnu dat bez nutnosti programování zajišťuje služba *RapidStart*. Zde uživatel založí konfigurační balíček přímo v BC, kde vybere objekt, který chce importovat / exportovat. Následně zavolá akci, která tuto výměnu provede. Následující obrázek obsahuje příklad balíčku pro import / export dat z tabulky zákazníka pomocí akcí *Export to Excel* a *Import from Excel*.

The screenshot shows the 'Config. Package Card' for a package named 'TEST'. The 'General' tab is active, displaying fields for Code (123), Language ID (0), Package Name, Processing Order (0), Product Version, and an 'Exclude Config. Tables' toggle. Below this is a 'Tables' section with a table listing the 'Customer' table. The table has columns for Table ID, Table Name, Parent Table ID, Data Template, and various statistics. The 'Customer' table is highlighted with a blue selection bar.

Table ID	Table Name	Parent Table ID	Data Template	Dim...	Skip	Del...	No. of Package	No. of Fields	No. of Fields	No. of Fields	No. of Package	No. of Database
				as	Table	Table	Records	Available	Included	To Validate	Errors	Records
				Col...	Trig...	Ref...						
18	Customer	0					0	107	107	106	0	392

Obrázek 3.1: Příklad definice konfiguračního balíčku, Zdroj: vlastní zpracování

Manipulace s daty touto cestou je většinou zdlouhavá a z pravidla se používá k prvotnímu nastavení databáze (import nastavení, migrace zákaznických dat, ...). Případně se používá k opravám chyb, které nastanou v průběhu používání systému.

3.2 Specifika oblasti

Výše zmíněné možnosti obstarávají samotný přenos dat po ručním vyvolání určité akce v systému. V rámci uživatelské přívětivosti nám zde chybí ale celá řada funkcí:

1. V *SaaS* není možnost automatického hromadného zpracování souborů v rámci lokálního úložiště.
2. Při výměně dat bychom chtěli sledovat počty přenosů dovnitř / ven ze systému. Chybí nám zde log pro jednotlivé transakce.

3. V rámci logu bychom chtěli rozlišovat komunikace pro jednotlivé typy zpracovávaných souborů (např. export objednávek, import karet zboží, ...).
4. V případě chyby přenosu potřebujeme chybu umět odchytit, zpracovat a opravit, aniž by zastavila zpracování dalších souborů.
5. Chyby je potřeba příslušnému uživateli zobrazit ideálně na domovské stránce BC, aby mohly být co nejdříve opraveny.
6. Při požadavku na výměnu dat pro nový typ souboru by mělo být relativně snadné tyto typy začlenit do již existujícího zpracování.

3.3 Cílové řešení

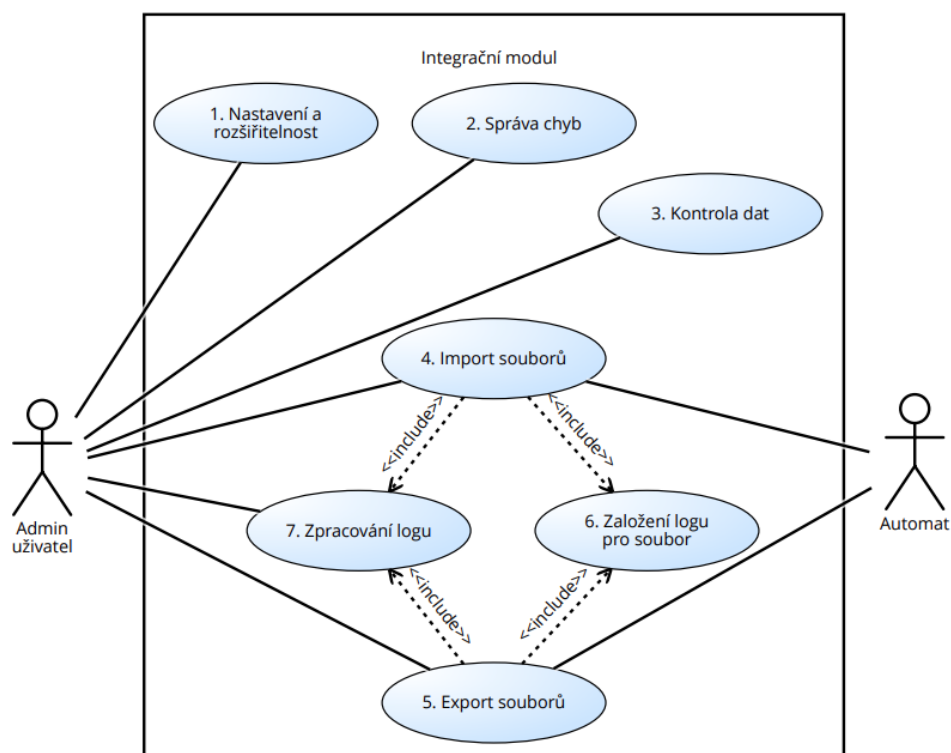
Na základě výčtu několika nedostatků standardního řešení v předchozí sekci se ve firmě rozhodlo, že bude vytvořeno nové univerzální rozšíření. Nový modul by měl být nezávislý na konkrétním zákazníkovi, aby bylo možné jej opakovaně nasazovat. Bude se starat čistě o výměnu dat mezi *SaaS* BC prostředím a lokálním úložištěm zákazníka. Měl by pokrýt všechny nedostatky zmíněné výše.

4 Návrh

Nyní podíváme na návrh klíčových funkcí nového modulu. Popíšeme si dva základní procesy importu a exportu souborů a v závěru této kapitoly si ukážeme návrh *class diagramu* nového modulu.

4.1 Funkční návrh

Na následujícím 4.1 *use case diagramu* můžeme vidět základní funkce modulu a aktéry, kteří je budou moci provádět.



Obrázek 4.1: Use Case diagram, Zdroj: vlastní zpracování

4.1.1 Use Case specifikace

Tato část obsahuje podrobnější popis jednotlivých *use case* z 4.1 diagramu.

1. Nastavení a rozšiřitelnost

Use Case: Nastavení a rozšiřitelnost
ID: 1
Popis: Nastavení integračního modulu v BC.
Primární aktéři: Admin uživatel
Sekundární aktéři: žádný
Předpoklady: 1. modul je nainstalovaný v BC
Hlavní tok: 1. Admin uživatel založí nastavení pro integrační modul. 2. Admin uživatel vyplní jednotlivá pole v nastavené správnými hodnotami. 3. Admin uživatel uloží nastavení.
Následné podmínky: 1 . Při správném nastavení probíhá komunikace se službou pro obsluhu souborů.
Alternativní tok: žádný

2. Správa chyb

Use Case: Správa chyb
ID: 2
Popis: Uživatelsky přívětivá správa chyb. Uživatel vidí chybovou hlášku s chybou, kterou má odstranit.

Primární aktéři:

Admin uživatel

Sekundární aktéři:

žádný

Předpoklady:

1. Existuje alespoň jedna položka, kterou se nepovedlo importovat / exportovat do / ze systému.

Hlavní tok:

1. Admin uživatel má přehled položek s chybou. 2. Na listu položek vyfiltruje admin uživatel položky s chybou.
 2. Dle chybových hlášek opraví admin uživatel data v systému.
 3. Admin uživatel ručně spustí akci na zpracování opravené položky.

Následné podmínky:

2. Neexistuje položka se stejnou chybou a data v systému jsou aktuální.

Alternativní tok:

žádný

3. Kontrola dat**Use Case:** Kontrola dat**ID:** 3**Popis:**

Oprávněný uživatel je schopný stáhnout importovaný soubor z BC.

Primární aktéři:

Admin uživatel

Sekundární aktéři:

žádný

Předpoklady:

1. Existuje položka v BC

Hlavní tok:

1. Admin uživatel si vyfiltruje požadovanou položku.
 2. Pomocí akce v BC si z položky stáhne soubor.

Následné podmínky:

3. Uživatel má k dispozici soubor, který je schopný otevřít a zkontrolovat jeho obsah.

Alternativní tok:

žádný

4. Import souborů**Use Case:** Import souborů

ID: 4

Popis:

Akce, která provede import souboru a jejich dat z daného úložiště do systému.

Primární aktéři:

Admin uživatel

Automat

Sekundární aktéři:

Webová služba

Předpoklady:

1. Existuje alespoň jeden soubor, který je potřeba importovat do BC.

Hlavní tok:

1. Admin uživatel / automat spustí akci na import souborů.

2. Akce založí položku pro každý soubor k importu.

3. Akce zpracuje nezpracované položky k importu.

Následné podmínky:

1. Existují zpracované položky typu import a data jsou naimportovaná v systému.

2. Existují položky s chybou importu a chyba je zapsána u dané položky.

Alternativní tok:

žádný

5. Export souborů

Use Case: Export souborů

ID: 5

Popis:

Akce, která provede export dat ve formě souboru pro dané záznamy ze systému do příslušného úložiště.

Primární aktéři:

Admin uživatel

Automat

Sekundární aktéři:

Webová služba

Předpoklady:

1. Existuje alespoň jeden záznam, který je potřeba exportovat z BC.

Hlavní tok:

1. Admin uživatel / automat spustí akci na export záznamu.

2. Akce založí položku pro každý záznam k exportu.

3. Akce zpracuje nezpracované položky k export.

Následné podmínky:

1. Existují zpracované položky typu export a data jsou exportovaná ze systému ve formě souboru.

2. Existují položky s chybou exportu a chyba je zapsána u dané položky.

Alternativní tok:

žádný

6. Založení logu pro soubor

Use Case: Založení logu pro soubor

ID: 6

Popis:

Akce, která pro daný typ souboru založí položku v BC.

Primární aktéři:

Admin uživatel

Automat

Sekundární aktéři:

žádný

Předpoklady:

1. Mám k dispozici soubor, pro který chci založit položku v BC a existuje CU pro zpracování souboru.

Hlavní tok:

1. Admin uživatel / automat spustí akci na založení položky.
2. Systém založí prázdnou položku s vygenerovaným primárním klíček.
3. Daná CU vyplní správně jednotlivé hodnoty položky a uloží změny.

Následné podmínky:

1. V systému existuje vyplněná položka obsahující příslušný soubor připravená ke zpracování.

Alternativní tok:

žádný

7. Zpracování logu**Use Case:** Zpracování logu**ID:** 7**Popis:**

Akce, která zpracuje nezpracované položky v BC.

Primární aktéři:

Admin uživatel

Automat

Sekundární aktéři:

Webová služba

Předpoklady:

1. V systému existuje alespoň jedna nezpracovaná položka a existuje CU na její zpracování.

Hlavní tok:

1. Admin uživatel / automat provede akci pro zpracování konkrétní položky.
2. Je spuštěna CU, která provede příslušné akce pro daný typ položky.

Následné podmínky:

1. Existuje zpracovaná položka v systému.
2. Existuje položka s chybou zpracování označená jako error s příslušnou chybovou hláškou.

Alternativní tok:

žádný

4.2 Procesní návrh

Dostáváme se k popisu dvou hlavních procesu – importu a exportu souborů a procesu jejich zpracování. Oba procesy mají podobný průběh, napřed dojde k založení položky protokolu integrace a poté je tato položka zpracována. Jsou zde znázorněny i případy, kdy proces neprobíhá ideální cestou.

4.2.1 Import souborů

Založení nastavení pro export

Uživatelská akce, která obnáší vyplnění nově založené tabulky v BC korektními daty pro export souborů.

Výpis souborů z adresáře

Systémová akce pro volání webové služby na práci se soubory s žádostí o výpis souborů v adresáři pro soubory k importu – dle nastavení v BC.

Import souboru do BC

Systémová akce pro stažení daného souboru do BC pomocí volání webové služby.

Založení protokolu integrace

Systémová akce, která na základě správně nastavené CU založí pro daný soubor položku protokolu integrace s korektně vyplněnými daty. Obsahuje samotný soubor určený k importu do BC.

Archivace souboru

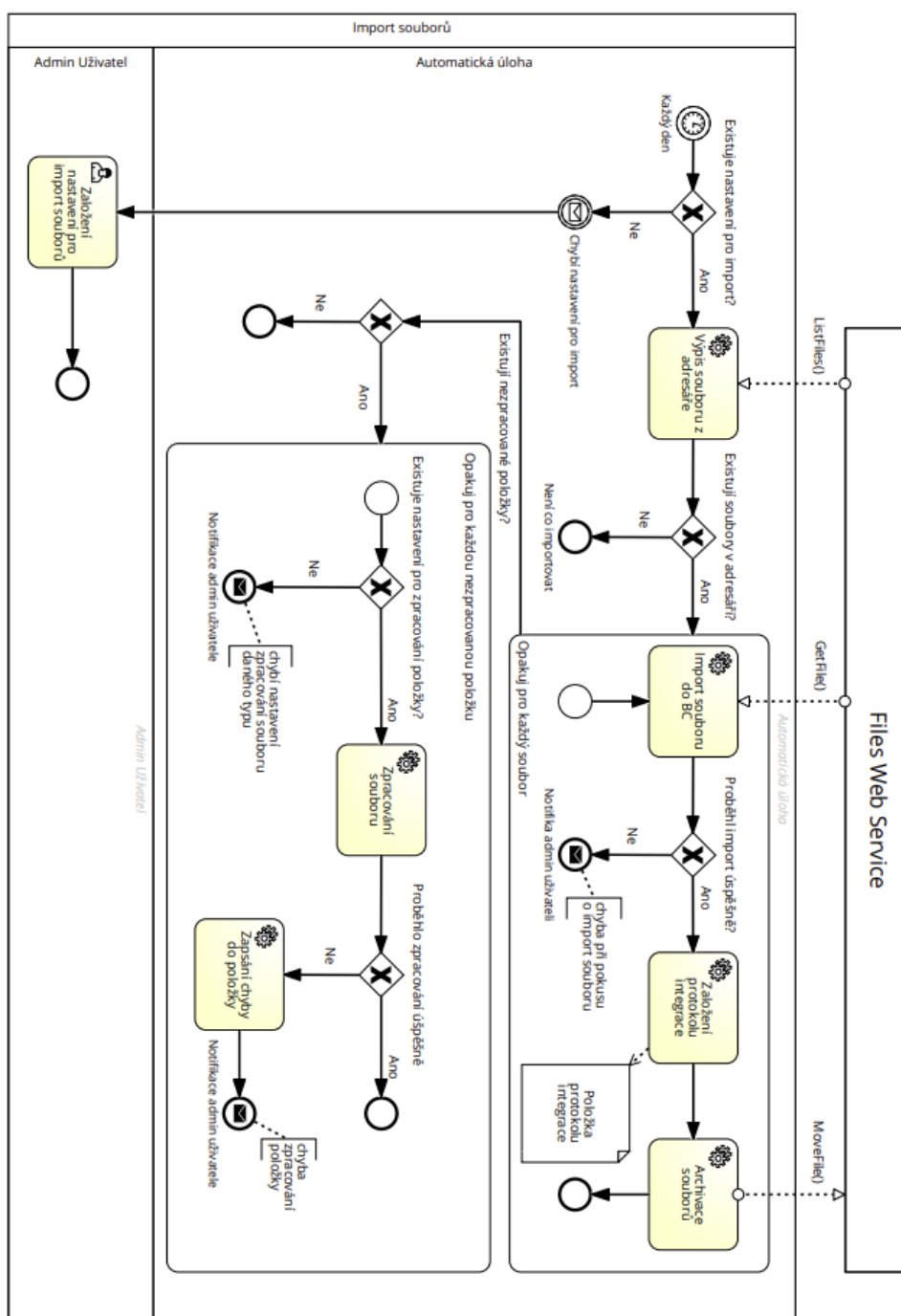
Systémová akce, která přesune úspěšně nainportovaný soubor pomocí volání webové služby do složky určené k archivaci souborů.

Zpracování souboru

Systémová akce na základě správně nastavené CU pro zpracování souboru daného typu integruje data určená k importu do BC dle pravidel v dané CU.

Zapsání chyby do položky

Systémová akce, která při chybě importu souboru označí položku protokolu integrace jako chybnou a zapíše k ní příslušnou chybovou hlášku.



Obrázek 4.2: Business Process Diagram - Import souborů, Zdroj: vlastní zpracování

4.2.2 Export souborů

Založení nastavení pro export

Uživatelská akce, která obnáší vyplnění nově založené tabulky v BC korektními daty pro export souborů.

Zpracuj záznam

Systémová akce na základě správně nastavené CU pro zpracování záznamu bude vytvořen soubor k exportu dle pravidel v dané CU.

Založení protokolu integrace

Systémová akce, která na základě správně nastavené CU založí položku protokolu integrace s korektně vyplněnými daty, která obsahuje soubor určený k exportu z BC.

Export souboru

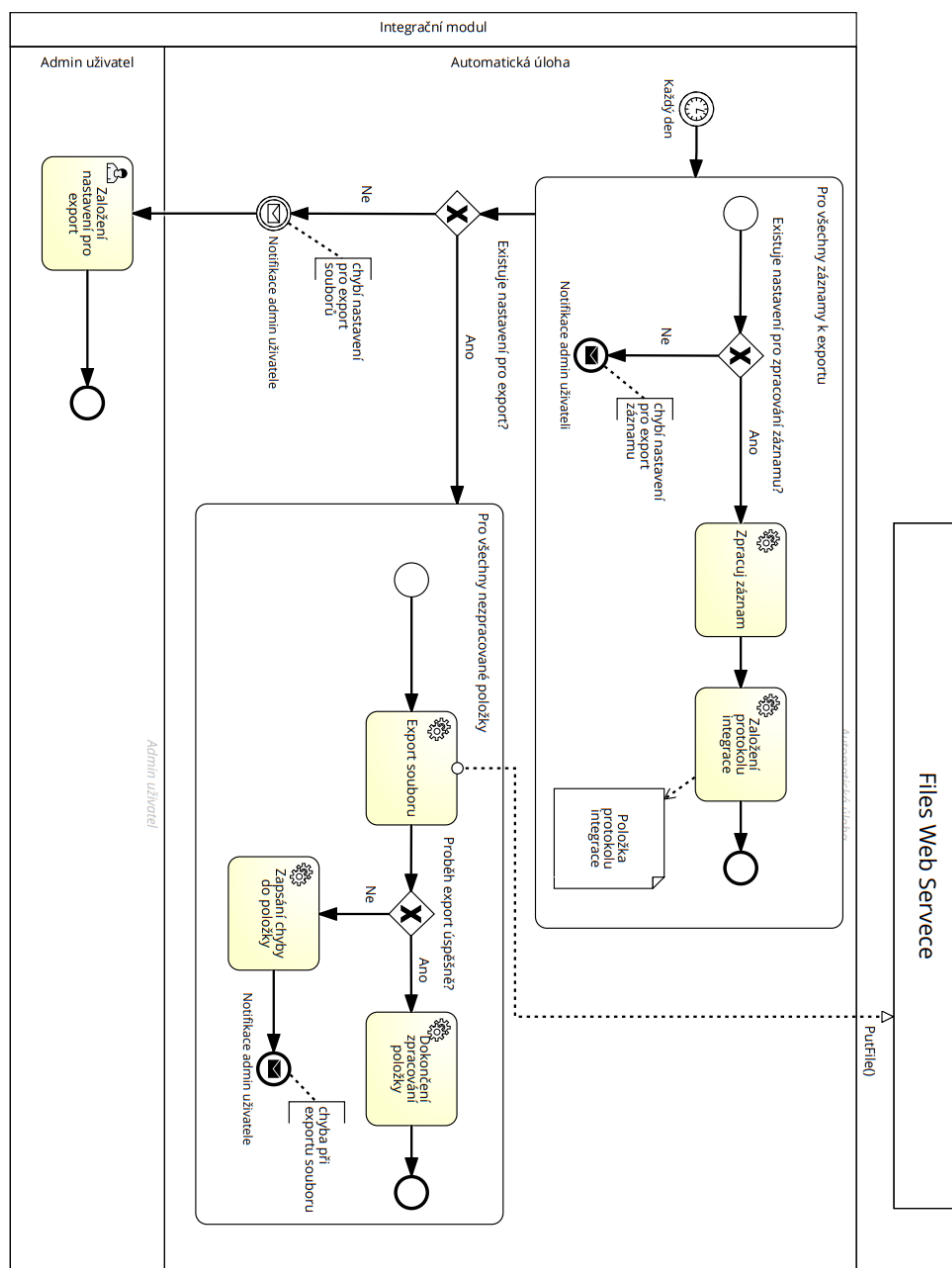
Systémová akce, která komunikuje s externí webovou službou pro správu souboru na daném zařízení. Provede samotný export nově vytvořeného souboru do lokálního úložiště uživatele dle nastavení v BC.

Dokončení zpracování položky

Po zpracování položky se označí příslušná položka jako zpracována a existuje v systému pouze jako záznam provedené akce.

Zapsání chyby do položky

Systémová akce, která při chybě exportu souboru označí položku protokolu integrace jako chybnou a zapíše k ní příslušnou chybovou hlášku.



Obrázek 4.3: Business Process Diagram - Export souborů, Zdroj: vlastní zpracování

4.3 Datový návrh

Součástí nové modulu bude několik nových objektů, bez kterých by nebylo možné splnit požadovanou funkcionalitu.

Communication Setup

Nová tabulka s poli pro nastavením celého modulu. Bude unikátní. Obsahuje pole pro nastavení URL adres sloužících ke komunikaci s webovou službou komunikující s lokálním úložištěm. Dále bude obsahovat masky příkazů pro práci se soubory.

Integration Log Entry

Nová tabulka uchovávající záznamy souborů, než dojde k jejich zpracování. Slouží současně jako přehled a kontrola historie přenosu dat. Obsahuje pole jako:

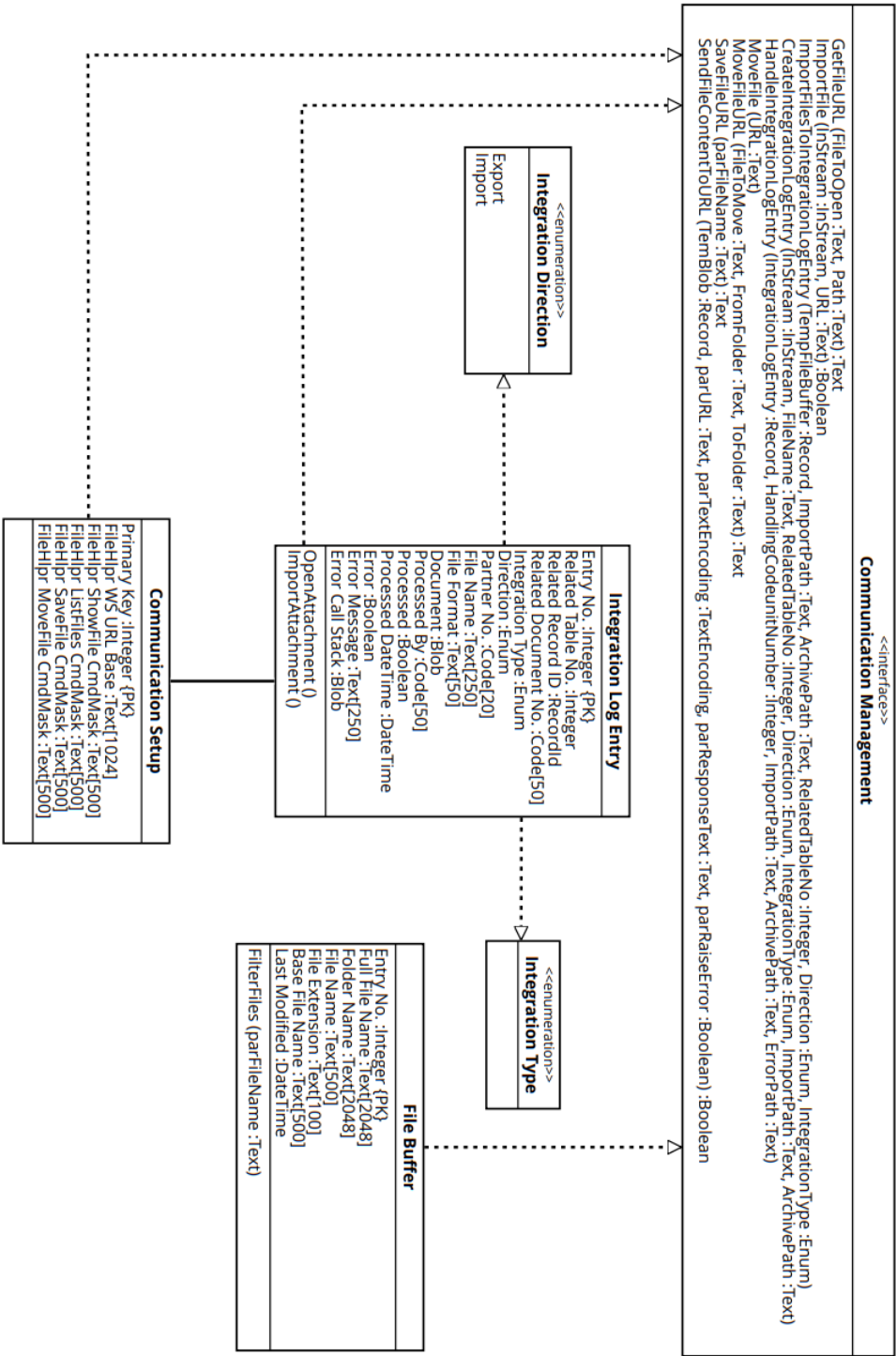
- číslo položky – primární klíč
- směr integrace – import / export
- typ integrace – pro různé typy souborů má různé zpracování
- informace o souboru – jméno, přípona, samotný soubor
- informace o související tabulce v BC
- informace o zpracování – zpracováno, chyba, chybová hláška, ...

File Buffer

Nová tabulka pro načtení souboru, která je předávána funkcím na následné zpracování. Obsahuje jeho základní informace.

Communication Management

Nová CU, která definuje funkce pro obsluhu jednotlivých souborů.



Obrázek 4.4: Class Diagram, Zdroj: vlastní zpracování 29

5 Implementace

V první řadě bylo založeno úložiště na gitu pro zálohování projektu. Dále byla založená kostra projektu a soubor s definicí základních parametrů nového modulu – unikátní id, název modulu, vydavatel, ...

Kód 5.1: Soubor s definicí projektu – app.json

```
1 {
2   "id": "88091073-c8a8-422c-b5db-bfca7b093038",
3   "name": "Essence Communication",
4   "publisher": "Essence Business Solutions",
5   "version": "21.0.0.0",
6   "brief": "The Essence Communication Project extension of the
7     Dynamics 365 Business Central (BC) brings an upgrade for
8     external communication.",
9   "description": "New file logging table and universal settings
10     for future extensions to specific formats.",
11   "privacyStatement": "https://essencebs.eu/en/privacy-policy",
12   "EULA": "https://essencebs.eu/en/software-app-end-user-
13     license-agreement",
14   "help": "https://docs.essencebs.cz/en-US/financials-base/
15     userguide/overview.html",
16   "url": "https://essencebs.eu/en",
17   "logo": "essence.png",
18   "contextSensitiveHelpUrl": "https://docs.essencebs.cz/{0}/
19     financials-base/userguide/",
20   "supportedLocales": [
21     "en-US", "cs-CZ"
22   ],
23   "dependencies": [],
24   "screenshots": [],
25   "internalsVisibleTo": [
26     {
27       "id": "a4f7779c-d82f-458f-9a0d-1c9770bdfef1c",
28       "name": "Essence Communication Tests",
29       "publisher": "Essence Business Solutions"
30     }
31   ],
32 }
```

```

26 "platform": "21.0.0.0",
27 "application": "21.0.0.0",
28 "idRanges": [
29   {
30     "from": 50100,
31     "to": 50149
32   }
33 ],
34 "resourceExposurePolicy": {
35   "allowDebugging": true,
36   "allowDownloadingSource": false,
37   "includeSourceInSymbolFile": false
38 },
39 "features": [
40   "TranslationFile",
41   "NoImplicitWith",
42   "TranslationFile",
43   "GenerateCaptions"
44 ],
45 "runtime": "10.0",
46 "target": "Cloud"
47 }

```

V programovacím jazyce AL Code byly následně vytvořeny jednotlivé objekty dle class digramu 4.4. To znamená, že byly vytvořeny tabulky pro ukládání dat a stránky pro grafické zobrazení v BC.

Nastavení komunikací

V prvním kole bylo založeno samotné nastavení modulu. Klíčová tabulka obsahující pole pro definici parametrů webové služby zajišťující přenos souborů mezi počítačem a BC společně s příkazy pro přesun souboru v rámci lokálního úložiště volané z BC. Stránka nastavení je typu karta a v systému existuje maximálně jeden záznam s nastavením.

Tyto objekty (tabulku a stránku) budou rozšiřovány v zákaznických modulech o parametry jednotlivých formátů pro integraci – cesty ke složkám v lokálním úložišti, maska jména souborů, formát souborů, číselné řady pro soubory, ...

Kód 5.2: Nastavení modulu – tabulka

```

1 table 50101 "Communication Setup EBS"
2 {
3     Caption = 'Communication Setup';
4     DataClassification = CustomerContent;
5     LookupPageId = "Communication Setup EBS";
6     DrillDownPageId = "Communication Setup EBS";
7
8     fields
9     {
10         field(1; "Primary Key"; Code[10])
11         {
12             Caption = 'Primary Key';
13         }
14         field(10; "FileHlpr WS URL Base"; Text[1024])
15         {
16             Caption = 'File Helper WS URL';
17         }
18         field(11; "FileHlpr ShowFile CmdMask"; Text[500])
19         {
20             Caption = 'Show file command mask';
21         }
22         field(12; "FileHlpr ListFiles CmdMask"; Text[500])
23         {
24             Caption = 'List files command mask';
25         }
26         field(13; "FileHlpr SaveFile CmdMask"; Text[500])
27         {
28             Caption = 'Save file command mask';
29         }
30         field(14; "FileHlpr MoveFile CmdMask"; Text[500])
31         {
32             Caption = 'Move file command mask';
33         }
34     }
35
36     keys
37     {
38         key(Key1; "Primary Key")
39         {

```



```

40         Clustered = true;
41     }
42 }
43 }

```

Kód 5.3: Nastavení modulu – stránka

```

1 page 50102 "Communication Setup EBS"
2 {
3     Caption = ' Communication Setup';
4     PageType = Card;
5     ApplicationArea = All;
6     UsageCategory = Administration;
7     SourceTable = "Communication Setup EBS";
8     DeleteAllowed = false;
9     InsertAllowed = false;
10    LinksAllowed = false;
11
12    layout
13    {
14        area(Content)
15        {
16            group(FileHelper)
17            {
18                Caption = 'File Helper';
19
20                field(FileHlprWSURLBase; Rec."FileHlpr WS URL
21                    Base")
22                {
23                    ToolTip = 'Specifies the file handling web
24                        service URL. Eq. http://192.168.10.123/
25                        FileProxy/filesystemproxy.svc/';
26                    ApplicationArea = All;
27                }
28                field(FileHlprShowFileCmdMask; Rec."FileHlpr
29                    ShowFile CmdMask")
30                {
31                    ToolTip = 'Specifies the show file command
32                        mask for web service URL. Eq. GetFile?
33                        user=%1&path=%2';

```

```

28         ApplicationArea = All;
29     }
30     field(FileHlprListFilesCmdMask; Rec."FileHlpr
31         ListFiles CmdMask")
32     {
33         ToolTip = 'Specifies the list files command
34             mask for web service URL. Eq. GetFileList
35             ?Path=%1&mask=%2&includesubdir=%3';
36         ApplicationArea = All;
37     }
38     field(FileHlprSaveFileCmdMask; Rec."FileHlpr
39         SaveFile CmdMask")
40     {
41         ToolTip = 'Specifies the save files command
42             mask for web service URL. Eq. PutFile?
43             Path=%1 Example: PutFile?Path=//
44             networkpath/folder/FileName.txt';
45         ApplicationArea = All;
46     }
47     field("FileHlpr MoveFile CmdMask"; Rec."FileHlpr
48         MoveFile CmdMask")
49     {
50         ApplicationArea = All;
51         ToolTip = 'Specifies the move file command
52             mask for web service URL. Eq. MoveFile?
53             User=%1&PathFrom=%2&PathTo=%3';
54     }
55 }
56
57 trigger OnOpenPage()
58 begin
59     Rec.Reset();
60     if not Rec.Get() then begin
61         Rec.Init();
62         Rec.Insert()
63     end;
64 end;
65 }

```

Položka protokolu integrace

Poté následovala implementace tabulky pro položku protokolu integrace 5.4 a stránky postavenou nad touto tabulkou 5.5. Důležité je, že položky není možné uživatelsky jakkoliv měnit ani mazat. Položky na stránce jsou seřazeny od nejnovější – dle narůstajícího čísla primárního klíče od největšího.

Kód 5.4: Položka protokolu integrace – tabulka

```

1 table 50100 "Integration Log Entry EBS"
2 {
3     Caption = 'Integration Log Entry';
4     DataClassification = CustomerContent;
5     LookupPageId = "Integration Log Entries EBS";
6     DrillDownPageId = "Integration Log Entries EBS";
7
8     fields
9     {
10         field(1; "Entry No."; Integer)
11         {
12             AutoIncrement = true;
13             Caption = 'Entry No.';
14             Editable = false;
15         }
16         field(10; "Related Table No."; Integer)
17         {
18             Caption = 'Related Table No.';
19             Editable = false;
20         }
21         field(20; "Related Record ID"; RecordId)
22         {
23             Caption = 'Related Record ID';
24             Editable = false;
25         }
26         field(30; "Related Document No."; Code[50])
27         {
28             Caption = 'Related Document No.';
29             Editable = false;
30         }
31     }
32 }
```

```

31     field(40; "Integration Type"; Enum "Integration Type
32         EBS")
33     {
34         Caption = 'Integration Type';
35     }
36     field(50; Direction; Enum "Integration Direction EBS")
37     {
38         Caption = 'Direction';
39     }
40     field(1000; "Partner No."; Code[20])
41     {
42         Caption = 'Partner No.';
43     }
44     field(1010; "File Name"; Text[250])
45     {
46         Caption = 'File Name';
47     }
48     field(1020; "File Format"; Text[50])
49     {
50         Caption = 'File Format';
51     }
52     field(1030; Document; Blob)
53     {
54         Caption = 'Document';
55     }
56     field(2000; "Processed By"; Code[50])
57     {
58         Caption = 'Processed By';
59         //TableRelation = User."User Name";
60     }
61     field(2010; Processed; Boolean)
62     {
63         Caption = 'Processed';
64
65         trigger OnValidate()
66         begin
67             if Processed then
68                 "Processed DateTime" := CurrentDateTime()
69             else
70                 "Processed DateTime" := ODT;

```

```
70         end;
71     }
72     field(2020; "Processed DateTime"; DateTime)
73     {
74         Caption = 'Processed DateTime';
75     }
76     field(2030; Error; Boolean)
77     {
78         Caption = 'Error';
79
80         trigger OnValidate()
81         begin
82             if not Error then
83                 "Error Message" := '';
84             end;
85         }
86     field(2040; "Error Message"; Text[250])
87     {
88         Caption = 'Error Message';
89     }
90     field(2050; "Error Call Stack"; Blob)
91     {
92         Caption = 'Error Call Stack';
93     }
94 }
95
96 keys
97 {
98     key(Key1; "Entry No.")
99     {
100     }
101     key(Key2; Direction, "Partner No.", Processed)
102     {
103     }
104     key(Key3; "Related Record ID")
105     {
106     }
107 }
108
109 fieldgroups
```

```
110 {
111 }
112
113 procedure OpenAttachment()
114 var
115     FileMgt: Codeunit "File Management";
116     TempBlob: Codeunit "Temp Blob";
117     Mask12Lbl: Label '%1.%2', Comment = '%1,%2';
118 begin
119     if not Document.HasValue() then
120         exit;
121     TempBlob.FromRecord(Rec, FieldNo(Document));
122     FileMgt.BLOBExport(TempBlob, StrSubstNo(Mask12Lbl, Rec
123         . "File Name", Rec. "File Format"), true);
124
125 procedure ImportAttachment()
126 var
127     FileMgt: Codeunit "File Management";
128     IStream: InStream;
129     OStream: OutStream;
130     FileName: Text;
131 begin
132     UploadIntoStream(SelectFileTxt, '', '', FileName,
133         IStream);
134     if FileName = '' then
135         exit;
136
137     if Document.HasValue() then
138         if not Confirm(ReplaceExistingFileQst) then
139             exit;
140
141     Clear(Document);
142     Document.CreateOutStream(OStream);
143     CopyStream(OStream, IStream);
144
145     "File Name" := CopyStr(FileMgt.
146         GetFileNameWithoutExtension(FileName), 1, MaxStrLen(
147         Rec. "File Name"));
```

```

145     "File Format" := CopyStr(FileMgt.GetExtension(FileName),
146                             1, MaxStrLen(Rec."File Format"));
147
148     Modify();
149 end;
150
151 var
152     ReplaceExistingFileQst: Label 'Do you want to replace
153     the existing file?';
154     SelectFileTxt: Label 'Select file';
155 }

```

Kód 5.5: Položka protokolu integrace – stránka

```

1 page 50101 "I. Log Entry Error FactBox EBS"
2 {
3     Caption = 'Last Error Information';
4     PageType = ListPart;
5     DeleteAllowed = false;
6     InsertAllowed = false;
7     ModifyAllowed = false;
8     LinksAllowed = false;
9     Editable = false;
10    SourceTable = "Integration Log Entry EBS";
11
12    layout
13    {
14        area(Content)
15        {
16            group(GroupName)
17            {
18                ShowCaption = false;
19
20                field(Error; Rec.Error)
21                {
22                    ApplicationArea = All;
23                    ToolTip = 'Specifies the value of the Error
24                        field.';
25                }
26            }
27        }
28    }
29 }

```

```

25         field("Error Message"; Rec."Error Message")
26         {
27             ApplicationArea = All;
28             ToolTip = 'Specifies the value of the Error
                Message field.';
29             MultiLine = true;
30         }
31         field("Error Call Stack"; ErrorCallStack)
32         {
33             ApplicationArea = All;
34             Caption = 'Error Call Stack';
35             ToolTip = 'Specifies the value of the Error
                Call Stack field.';
36             MultiLine = true;
37         }
38     }
39 }
40 }
41
42 trigger OnAfterGetCurrRecord()
43 var
44     CallStackInStream: InStream;
45 begin
46     Rec.CalcFields("Error Call Stack");
47     Rec."Error Call Stack".CreateInStream(CallStackInStream
        );
48     CallStackInStream.ReadText(ErrorCallStack);
49 end;
50
51 var
52     ErrorCallStack: Text;
53 }

```

Typ integrace

V rámci implementace byl vytvořen objekt typu enum 5.6, který je prázdný. To z toho důvodu, že je potřeba se na něj odkazovat v modulu, ale není potřeba znát jednotlivé hodnoty. Je nachystán pro budoucí

rozšíření o konkrétní typy souborů požadovaných pro integraci v jednotlivých zákaznických modulech.

Kód 5.6: Typ integrace – enum

```
1 enum 50101 "Integration Type EBS"
2 {
3     Extensible = false;
4
5     value(0; " ")
6     {
7         Caption = ' ', Locked = true;
8     }
9 }
```

Práva modulu

Dále bylo třeba celému modulu nastavit práva, na jednotlivé objekty, která lze následně přiřadit koncovým uživatelům tohoto modulu. Standardně se vytváří dvě sady práv a to jedna pouze pro zobrazování objektů 5.7 a druhá pro celkovou editaci 5.8.

Dříve byly práva generována pomocí XML souborů, nově jsou to samostatné objekty.

Kód 5.7: Práva k modulu – READ

```
1 permissionset 50100 "Communic.-READ EBS"
2 {
3     Assignable = true;
4     Permissions =
5         tabledata "Communication Setup EBS" = R,
6         tabledata "File Buffer EBS" = R,
7         tabledata "Integration Log Entry EBS" = R,
8         tabledata "Communication Cue EBS" = R;
9 }
```

Kód 5.8: Práva k modulu – EDIT

```
1 permissionset 50101 "Communic.-EDIT EBS"
2 {
3     Assignable = true;
4     IncludedPermissionSets = "Communic.-READ EBS";
5
6     Permissions =
7         tabledata "Communication Setup EBS" = IMD,
8         tabledata "File Buffer EBS" = IMD,
9         tabledata "Integration Log Entry EBS" = IMD,
10        tabledata "Communication Cue EBS" = IMD;
11 }
```

6 Testování

Testování probíhalo pouze ručním spouštěním. Byl vytvořen BC Sandbox¹, kde probíhalo ladění komunikace ve spolupráci s kolegou, který vytvářel webovou službu zajišťující správu souborů.

V první fázi jsme ladili samotné nastavení BC, aby bylo možné zavolat http request se správnými parametry. Ladil se odkaz na webovou službu a jednotlivé příkazy, které je možné pomocí této služby volat. Bylo potřeba správně vyplnit novou tabulku nastavení a masky pro potřebné příkazy. V rámci této fáze se také testovaly funkce na volání http příkazů a to konkrétně na získání souboru, na přesun souboru a na uložení nového souborů.



Obrázek 6.1: Webová služba – přehled příkazů, Zdroj: vlastní zpracování

V druhé fázi se ladilo samotné zakládání protokolu integrace.to probíhalo již v kombinaci se zákaznickým rozšířením o konkrétní formát importovaného souboru. Nejdříve se ladilo správné založení položky na základě jednoho ručně importovaného souborů. Bylo potřeba zajistit správné číslování položek. Později byla vytvořena úloha na automatické zpracování tohoto formátu souboru.

1. Sandbox – bezpečné online prostředí určené pro testování, trénink, ...
<https://learn.microsoft.com/en-us/dynamics365/business-central/admin-sandbox-environments>

6.1 Problémy objevené při testování

6.1.1 Zpracování všech souborů v adresáři

V průběhu přišel požadavek, aby byly zpracovány všechny soubory adresáře. Původně se zpracování zastavovalo na prvním chybném souboru. Zde bylo potřeba správně ošetřit chybný soubor, zapsat správně errorové hlášky do dané položky. Toho bylo dosaženo jednoduchým trikem – podmíněné spuštění CU na obsluhu dané položky.

```
internal procedure HandleIntegrationLogEntry(var IntegrationLogEntry: Record "Integration Log Entry EBS"; HandlingCodeunitNumber: Integer)
var
    IntegrationLogEntry2: Record "Integration Log Entry EBS";
    OutStr: OutStream;
begin
    IntegrationLogEntry.SetAutoCalcFields(Document);
    if IntegrationLogEntry.FindSet() then
        repeat
            IntegrationLogEntry2 := IntegrationLogEntry;
            Commit();

            if not Codeunit.Run(HandlingCodeunitNumber, IntegrationLogEntry2) then begin
                IntegrationLogEntry2.Error := true;
                IntegrationLogEntry2."Error Message" := CopyStr(GetLastErrorText(), 1, MaxStrLen(IntegrationLogEntry2."Error Message"));
                IntegrationLogEntry2."Error Call Stack".CreateOutStream(OutStr);
                OutStr.WriteText(GetLastErrorCallStack());

                IntegrationLogEntry2.Modify();
            end else begin
                IntegrationLogEntry2.Processed := true;
                IntegrationLogEntry2."Processed By" := CopyStr(UserId(), 1, MaxStrLen(IntegrationLogEntry2."Processed By"));
                IntegrationLogEntry2."Processed DateTime" := CurrentDateTime();
                IntegrationLogEntry2.Modify();
            end;
        until IntegrationLogEntry.Next() = 0;
    end;
end;
```

Obrázek 6.2: Podmínění spuštění CU, Zdroj: vlastní zpracování

6.1.2 Duplicitní importované soubory

V počátku docházelo k přesunu souboru do archivní složky až po obsluhu daného souboru, nicméně v případě neočekávané chyby k tomuto přesunu nikdy nedošlo. Proto byla k tomuto souboru v dalším kole importu vytvořena nová položka v integračním protokolu. Tato chyba byla opravena přesunem archivace ihned po importu souboru do položek v BC a až poté následovalo zpracování této položky.

```
internal procedure CreateIntegrationLogEntry(var InStream: InStream; FileName: Text; RelatedTableNo: Integer; Direction: Enum "Integration Direction EBS";
var
    IntegrationLogEntry: Record "Integration Log Entry EBS";
    FileManagement: Codeunit "File Management";
    OutStream: OutStream;
begin
    Clear(IntegrationLogEntry);
    IntegrationLogEntry.Init();
    IntegrationLogEntry."Related Table No." := RelatedTableNo;
    IntegrationLogEntry.Direction := Direction;
    IntegrationLogEntry."Integration Type" := IntegrationType;

    Clear(IntegrationLogEntry.Document);
    IntegrationLogEntry.Document.CreateOutStream(OutStream);
    CopyStream(OutStream, InStream);

    IntegrationLogEntry."File Name" := CopyStr(FileManagement.GetFileNameWithoutExtension(FileName), 1, MaxStrLen(IntegrationLogEntry."File Name"));
    IntegrationLogEntry."File Format" := CopyStr(FileManagement.GetExtension(FileName), 1, MaxStrLen(IntegrationLogEntry."File Format"));
    IntegrationLogEntry.Insert();

    if (ImportPath <> '') and (ArchivePath <> '') then
        MoveFile(MoveFileURL(FileName, ImportPath, ArchivePath)); // to avoid multiple imports in case of process file fail
end;
```

Obrázek 6.3: Přesun souboru – akce pro založení položky integrace,
Zdroj: vlastní zpracování

6.1.3 Ošetření ručního importu souboru

V tomto případě neznáme výchozí cestu k souboru a funkce na import padala ve fázi pokusu o přesun souboru, kde se snažila volat přesun z nedefinované cesty. Toto chování bylo ošetřeno jednoduchým dotazem na prázdnotu parametru viz 6.3.

7 Provoz

V současné chvíli modul využívá v ostrém provozu jeden zákazník. U dalších dvou zákazníků je tento modul plánován na integraci, ale implementace zatím probíhá v jiných kritičtějších oblastech.

V rámci této kapitoly si ukážeme, jak mohou vypadat jednotlivé objekty v ostrém provozu.

7.1 Nastavení komunikací

Na následujícím obrázku můžeme vidět, jak vypadá stránka s nastavením modulu pro komunikace. Můžeme vidět, že byla tato tabulka rozšířena o nová pole týkající se specifikace jednotlivých typů souborů. Především se jedná o cesty k importovaným souborům, dále pak o cesty k archivaci souborů. Důležité jsou i číselné řady, které zajistí založení záznamu v BC s vhodným primárním klíčem.

The screenshot shows the 'Communication Setup' page in Dynamics 365 Business Central. The page is divided into two main sections: 'File Helper' and 'EDI'. The 'File Helper' section contains fields for 'File Helper WS URL', 'FileHelper User ID', 'FileHelper User Pass', 'List files command mask', 'Show file command mask', 'Save file command mask', 'Move file command mask', 'Delete file command mask', and 'Execute command mask'. The 'EDI' section contains fields for 'EDI Import File Path', 'EDI Archive File Path', 'EDI Error File Path', 'EDI ORDERS Nosk', 'Directory for Export EDI INV', 'Directory for Export EDI DES', 'Dir. for Export EDI INV - Seeburger', 'Directory for Export EDI DES - Seeburger', 'Dir. for Export EDI DES - Seeburger', 'Directory for Export EDI PO', 'No. Series for EDI File EDI', 'No. Series File Order EDI', 'File Extension EDI', 'TKL Daily Aut. Shipping', 'Automatic CSV Handling', and 'Automatic Order Creation'. There are also dropdown menus for 'EDI ORDERS Nosk', 'File Extension EDI', and 'No. Series for EDI File EDI'. The page has a top navigation bar with 'Dynamics 365 Business Central' and a bottom section with 'BarTender' and 'BridgeWeight'.

Obrázek 7.1: Nastavení komunikací – reálný příklad, Zdroj: vlastní zpracování

7.2 Položka protokolu integrace

Zde můžeme vidět položky protokolu integrace, které byly úspěšně zpracovány. V případě chyby by bylo vidět na bočním panelu přesný popis. Dále si na následujícím obrázku můžeme všimnout jednotlivých typů integrovaným položek – došlo k rozšíření základního objektu typu enum ?? o konkrétní typy.

Entry No.	Integration Type	Direction	Related Exported Document No.	Related Imported EDI Orders	Related Unprocessed EDI Orders	Related Table No.	Partner No.	File Name
222021	SivaPlan	Import	...	0	0	56630	...	L230426224658
222020	SivaPlan	Import	...	0	0	56630	...	L230426224757
222019	SivaPlan	Import	...	0	0	56630	...	L230426224656
222018	SivaPlan	Import	...	0	0	56630	...	L230426224556
222017	SivaPlan	Import	...	0	0	56630	...	L230426224455
222016	SivaPlan	Import	...	0	0	56630	...	L230426224354
222015	SivaPlan	Import	...	0	0	56630	...	L230426224253
222014	SivaPlan	Import	...	0	0	56630	...	L230426224153
222013	BWeight	Import	...	0	0	56600	...	EM_TARA_2023_04_26_...
222012	SivaPlan	Import	...	0	0	56630	...	L230426224052
222011	SivaPlan	Import	...	0	0	56630	...	L230426223947
222010	SivaPlan	Import	...	0	0	56630	...	L230426223846
222009	SivaPlan	Import	...	0	0	56630	...	L230426223746
222008	SivaPlan	Import	...	0	0	56630	...	L230426223645
222007	SivaPlan	Import	...	0	0	56630	...	L230426223544
222006	BWeight	Import	...	0	0	56600	...	EM_ABRUPPEN_2023_04_...
222005	SivaPlan	Import	...	0	0	56630	...	L230426223443
222004	SivaPlan	Import	...	0	0	56630	...	L230426223342
222003	SivaPlan	Import	...	0	0	56630	...	L230426223242
222002	SivaPlan	Import	...	0	0	56630	...	L230426223141
222001	SivaPlan	Import	...	0	0	56630	...	L230426223040

Obrázek 7.2: Položka protokolu integrace – reálný příklad, Zdroj: vlastní zpracování

7.3 Automatické úlohy

Na následujícím příkladu vidíme automaticky spouštěné úlohy na obsluhu integrovaných souborů. Na základě uživatelského nastavení BC automaticky pouští úlohu po uplynutí daného časového intervalu.

7. PROVOZ

Dynamics 365 Business Central

Job Queue Entries

Search + New Edit Last Delete Edit View Set Status to Ready Restart Run once (foreground) Set On Hold Show Error More options

Status	User ID	Object Type to Run	Object ID to Run	Object Caption to Run	Description	Job Queue Category Code	User Session Started	Earliest Start Date/Time	Sched. Job	Recall Job	No. of Minutes between Runs
<input type="radio"/> Error		Codeunit	6700	C365 Sync. Management				01.08.2022 08:17	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Error		Codeunit	1441	RC Headlines Executor				01.08.2022 08:21	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="radio"/> Error		Codeunit	3997	Retention Policy JG		RETENTION		06.09.2022 02:00	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="radio"/> Error		Codeunit	1441	RC Headlines Executor				02.11.2022 14:09	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="radio"/> Ready		Report	1511	Delegate Approval Requests				27.04.2023 10:12	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Error		Codeunit	8811	Customer Statement via Queue			17.02.2023 08:26	17.02.2023 08:26	<input type="checkbox"/>	<input type="checkbox"/>	0
<input type="radio"/> Ready		Codeunit	56640	Auto Production Planning				02.05.2023 00:00	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input checked="" type="radio"/> Ready		Codeunit	56635	SP Streaming XMLport				27.04.2023 06:55	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	60
<input checked="" type="radio"/> Ready		Codeunit	56651	Import EDI Files				26.04.2023 23:04	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10
<input checked="" type="radio"/> Ready		Codeunit	56652	Export EDI Files				26.04.2023 23:00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10
<input type="radio"/> Error		Codeunit	56606	Forecast Summarized			18.04.2023 00:00	18.04.2023 00:00	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Error		Codeunit	56606	Forecast Summarized			17.04.2023 00:00	17.04.2023 00:00	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Error		Codeunit	56606	Forecast Summarized			17.04.2023 00:00	17.04.2023 00:00	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Ready		Codeunit	56606	Forecast Summarized				27.04.2023 00:00	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Ready		Codeunit	56608	Post Adjustment				27.04.2023 21:13	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Ready		Codeunit	56606	Forecast Summarized				27.04.2023 00:01	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input type="radio"/> Error		Codeunit	56606	Forecast Summarized			24.04.2023 00:00	24.04.2023 00:00	<input type="checkbox"/>	<input type="checkbox"/>	1440
<input checked="" type="radio"/> Ready		Codeunit	56639	Siva Droid Runner				26.04.2023 22:59	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2
<input checked="" type="radio"/> Ready		Codeunit	56600	Import Weights Barzotec				26.04.2023 23:01	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5
<input type="radio"/> Ready		Codeunit	56646	Automated Printing				02.05.2023 00:00	<input type="checkbox"/>	<input type="checkbox"/>	1440

Obrázek 7.3: Položka protokolu integrace – reálný příklad, Zdroj: vlastní zpracování

8 Vyhodnocení

Celkově si myslím, že se modul podařilo úspěšně implementovat. Aktuálně běží v živém provozu u jednoho zákazníka. Ten byl tak laskav a poskytl mi na modul zpětnou vazbu, která proběhla formou vyplnění dotazníku prostřednictvím emailové komunikace.

Zpětnou vazbu bych chtěla provádět po každém významnějším vylepšení, abych mohla sbírat data na případná další rozšíření.

8.1 Zpětná vazba

Ráda bych upozornila, že u tohoto zákazníka došlo k rozšíření základního modulu o jednotlivé formáty k integraci, proto v dotazníku zmiňuje již konkrétní typu integrace jako např. automatické sklady, EDI, ...

1. Jaká je poruchovost modulu?

(a) Jak často řešíte problémy tohoto modulu?

Hodně záleží na počtu a kvalitě zpracovávaných souborů, které občas vykazují chyby, které zpracovávanou funkci zastaví, např.: Komunikace s automatickým skladem – probíhá oboustranně a podle testů je stabilní, pokud soubory načítáme každé 2 minuty, při opakovaném načtení po 1 minutě se často načítání zablokovalo a při následném automatickém pokusu o spuštění se již načtený soubor importoval znovu, což zablokovalo další pokusy. Takové položky v Intergration Logu bylo třeba ručně označit jako zpracované a službu spustit znovu. Při prodloužení intervalu na 2 minuty se tento problém přestal vyskytovat. Modul Communication je celkově stabilní.

(b) Jste schopný problémy vyřešit pomocí přiložených nástrojů?

Ano, importované chybné soubory lze následně dobře opravit a dokončit import.

(c) Jak lehce lze identifikovat oblast problému dané integrace?

Velmi snadno, uložil jsem si filtry podle Intergration type a sloupce Processed=Ano - a tak jedním klikem jsem schopný zkontrolovat pokud někde nastala chyba a kde. Vzhledem k počtu

akcí, které modul zpracovává, by bylo komfortnější, kdyby modul při výskytu chyby při zpracování nějaké funkce zaslal e-mail - stejně jako máme nadefinované v Job Queue. Administrátor pak okamžitě o problému ví. Dnes např. problém při importu EDI souboru poznám podle adresáře Error, kam Intergration log nezpracovaný soubor přesune.

2. Jaký je průtok dat tohoto modulu?

- (a) Omezujete nějak objem integrovaných dat?
Ne, v Logu je něco přes 2 000 položek za 24 hodin.
- (b) Jaké jsou zpoždění ve zpracování?
 - i. Registrujete nějaká?
Zpracování nastává dle nastaveného času a opakování, ke zpoždění nedochází.
 - ii. Je to omezující?
Ne.

3. Jaká je spolehlivost modulu?

- (a) Víte o incidentu, kdy by došlo ke ztrátě dat?
Již popsáno výše, je třeba lépe informovat administrátora např. e-mailem.

4. Pokryl tento modul potřebné oblasti pro import?

- (a) Importujete stále nějaké soubory ručně?
Ne, pokud jsou soubory v patřičné kvalitě.
 - i. Šlo by tyto importy automatizovat v rámci tohoto modulu?
Máme speciální soubory Edifactu, kde v jednom EDI souboru je několik desítek prodejních objednávek určených pro import do BC, pokud v takovém souboru nastane při importu chyba (např. proto, že chybí EAN kód výrobku nebo daný EAN kód nebyl v systému nalezen), potom několik prodejních objednávek je nainportováno a zbylé zůstanou v Intergration logu nezpracované - tu jednu chybnou je třeba

opravit (např. doplnit EAN kód do systému) a ručně odklikat i ostatní jednu po druhé – vytvořit z nich objednávky – někdy jich je hodně.

5. Pracují s modulem i koncoví uživatelé?

(a) Například při opravách chyb?

Ano, díky modulu už nemusím všechny soubory importovat já, ale s mnoha chybami si dokáže uživatel poradit, je tu i zastupitelnost při dovozených atd.

6. V případě požadavku na import nového souboru:

(a) Myslíte si, že byste dokázal zadat požadavek na rozšíření o nový formát importu / exportu na základě stávajících zkušeností? (jaké tabulky rozšířit, akce, ...)

Určitě ano, v tuto chvíli jen pro import souborů Edifactu máme 8 typů, každý typ má svá specifika, která bylo třeba na začátku popsat.

i. Jak si myslíte, že se v tomto modulu orientujete?

Ze začátku to bylo trochu komplikované, nic podobného jsem dosud nepoužíval, nyní se v tom myslím dobře orientuji.

ii. Vyhovuje Vám konstrukce řešení? Určitě ano, ocenil jsem i prolínání s importovanými objednávkami, kdy z objednávky se lze prokliknout zpět na konkrétní položku v Logu.

iii. Jedná se o dost univerzální modul pro případné rozšíření?

Myslím, že plně pokryje veškerou komunikaci s externími systémy.

7. Máte nějaké další náměty na rozšíření tohoto modulu?

(a) Vyhovuje Vám design?

Plně vyhovuje.

(b) Ocenil byste nějaké nové prvky?

Kromě výše popsanych snad ještě možnost administrátorsky nastavit omezení pro uživatele na konkrétní Intergration type – pro účetní, která fakturuje Edifactové objednávky, jsem nastavil přístup na modul, kde vidí všechny položky, sama si nastavila filtr

na typ EDIFACT, ale má dostatek práv na to, aby modifikovala jiné položky – např. označila některé již zpracované položky jako nezpracované a tím se znovu načetly.

Jak vyplývá z této zpětné vazby, zákazník je s modulem spokojen. Modul se chová stabilně a nevykazuje známky větší poruchovosti, přestože zpracuje dle slov zákazníka něco přes 2000 položek za den. Nedochází ke zpožděním, díky automaticky nastavitelným úlohám. V případě chyby je zákazník s přiloženými nástroji chybu opravit a docílit tak úspěšné integrace.

Zákazník dále oceňuje, že s modulem mohou pracovat i koncoví uživatelé a dochází tak k zastupitelnosti v případě např. dovolených. Po určité době praxe se zákazník v tomto systému dobře orientuje, má potřebné prokliky na provázané doklady.

Díky zpětné vazbě jsem dostala i několik podnětů pro zlepšení tohoto modulu. Kromě filtrování na chybné položky lze modul vylepšit přidáním notifikace emailem při chybném zpracování položky. Tato úprava by pak nevyžadovala otevírat BC a kontrolovat zpracování.

Dále by pak zákazník ocenil možnost detailnější specifikace práv uživatelů v rámci tohoto modulu. To se dělá například přidáním polí typu boolean na tabulku nastavení uživatele, kde se povolí modifikace na jednotlivé typy integrace. Tento požadavek nejde zobecnit pro potřeby modulu a jedná se o zákaznické rozšíření na základě specifikace typu integrace.

Závěr

ERP systémy mají zastoupení většině firem. Pomáhají jim uchovávat data, optimalizovat procesy a tím i udržovat krok s konkurencí. Díky variabilitě trhu, si firmy mohou vybrat z mnoha druhů ERP, ať už krabicových řešení nebo investovat do systému na míru. Je potřeba počítat, že krabicové řešení nepokryje všechny potřeby firmy a bude potřeba investice do customizace systému.

Právě jednou takovou customizací se zabývala tato práce. Popsali jsme si zde na konkrétním příkladu ERP systému – BC, jak probíhá vývoj jednoho modulu. Řekli jsme si něco málo k teorii tohoto systému, podívali jsme se na jednotlivé objekty a na základě těchto znalostí jsme mohli začít proces vývoje nového modulu.

Po definici oblasti problému, který nastal v rámci implementace u zákazníka jsme analyzovali problém a došli jsme k návrhu konkrétního řešení. Z této části nám vyplynulo několik diagramů, které názorně zobrazovali cílové objekty a chování systému.

Následná implementace kopírovala návrh řešení a popisovala možnosti rozšíření dle požadavků zákazníka. Testování již probíhalo s částečně modifikovaným řešením o zákaznická data. V části provoz jsem ukázala reálné příklady z živého provozu a v části hodnocení mi byla poskytnuta v celku pozitivní zpětná vazba na tento modul.

Díky této práci jsem mohla aplikovat znalosti projektového řízení, tvorby diagramů a v neposlední řadě samotné implementace nového modulu. Rozšířila jsem si znalosti v programování http komunikace pomocí programovacího jazyku AL Code.

Tato práce mi ukázala, jak komplikované může být vytvoření obecného modulu, který vznikl na základě analýzy požadavků jednoho zákazníka. Nicméně díky zpětné vazbě věřím, že se mi podařilo vytvořit dostatečně obecné rozšíření užitečné pro širokou skupinu cílových zákazníků. Zároveň mi zpětná vazba poskytla náměty na budoucí rozvoj modulu a věřím, že díky opakovanému nasazení u více zákazníků se bude modul pouze zdokonalovat.

Bibliografie

1. NICOLÀS LORENTE, Laura; NICOLÀS LORENTE, Cristina. *Implementing Microsoft Dynamics NAV 2013*. 1st ed. Birmingham: Packt Publishing, 2013. ISBN 978-1-849686-02-0.
2. *Plánování podnikových zdrojů* [online]. Wikipedie, otevřená encyklopedie, 2022 [cit. 2023-03-03]. Dostupné z: https://cs.wikipedia.org/wiki/Pl%C3%A1nov%C3%A1n%C3%AD_podnikov%C3%BDch_zdroj%C5%AF.
3. *Podnikové informační systémy* [online]. Brno: Mgr. Jan Matula, PhD., 2015 [cit. 2023-03-03]. Dostupné z: https://is.muni.cz/el/phil/podzim2015/VIKBA18/um/VIKBA18_-_3._blok_ZS_2015.pdf.
4. *How to Manage ERP Customizations* [online]. Elizabeth Quirk, 2020-03-02 [cit. 2023-04-20]. Dostupné z: <https://solutionsreview.com/enterprise-resource-planning/how-to-manage-erp-customizations/>.
5. *ERP System Customizations: A Necessary Evil?* [online]. Carter Lloyds, 2017-09-01 [cit. 2023-04-20]. Dostupné z: <https://www.qad.com/blog/2017/09/erp-system-customizations>.
6. DEMILIANI, Stefano; TACCONI, Duilio. *Dynamics 365 Business Central Development Quick Start Guide*. 1st ed. Packt Publishing, 2018. ISBN 978-1-78934-746-3.
7. *ERP systém Dynamics 365 Business Central* [online]. Essence International s.r.o., 2023 [cit. 2023-04-21]. Dostupné z: <https://essencebs.eu/cz/erp-system-dynamics-365-business-central>.
8. *Extensibility and Collaboration in Business Central* [online]. Aleksandar Totovic, 2020-10-08 [cit. 2023-04-21]. Dostupné z: <https://totovic.com/2020/10/08/extensibility-and-collaboration-in-business-central/>.
9. BRUMMEL, Marije; STUDEBAKER, David; STUDEBAKER, Chris. *Programming Microsoft Dynamics 365 Business Central*. 6th ed. Packt Publishing, 2019. ISBN 978-1-78913-779-8.

A Přílohy

Součástí této práce je archivovaná složka s vypracovaným modulem nahraná:

- *prod-essence-communication-bc.zip* – zkompilevatelné řešení