

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Rozšíření integračního rozhraní pro plánovací systém

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 2. května 2019

Zdeněk Častorál

Poděkování

Tímto bych rád poděkoval svému vedoucímu bakalářské práce Ing. Martinu Hronovi za odborné vedení, za pomoc a cenné rady při zpracování této práce. Dále bych chtěl poděkovat svému konzultantovi bakalářské práce Ing. Romanu Moučkovi, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích této práce.

Abstract

The subject of this bachelor's thesis is integration of information systems. The aim of this paper is to extend an existing integration interface between the Enterprise Resource Planning (ERP) and Advanced Planning and Scheduling (APS) systems, which is being developed by the company AIMTEC a.s. This integration interface uses the relational database management system Microsoft SQL Server and is implemented in the T-SQL language.

This paper opens with an overview of the general issues surrounding information systems integration. It then moves on to focus on the ERP and APS systems. The next section of the paper analyses the existing integration interface, followed by a definition of the requirements placed on an extended integration interface. This includes their detailed specifications, a proposal for extending this integration interface with new functions, and the implementation technology. The conclusion describes testing of the implemented solution, including putting the extended integration interface into trial operation.

Abstrakt

Tato bakalářská práce se zabývá problematikou integrace informačních systémů. Cílem práce je rozšířit stávající integrační rozhraní mezi systémy Enterprise Resource Planning (ERP) a Advanced Planning and Scheduling (APS), které je vyvíjeno ve společnosti AIMTEC a.s. Toto integrační rozhraní využívá relační systém řízení báze dat Microsoft SQL Server a je implementováno v jazyce T-SQL.

Práce nejprve shrnuje obecnou problematiku integrace informačních systémů, dále se zaměřuje na systémy ERP a APS. V další části práce je analyzováno stávající integrační rozhraní. Poté jsou definovány požadavky na rozšířené integrační rozhraní včetně jejich detailní specifikace, návrh nových rozšiřujících funkcí tohoto integračního rozhraní a technologie použité k implementaci. V závěru je popsáno testování realizovaného řešení včetně nasazení rozšířeného integračního rozhraní do testovacího provozu.

Obsah

1	Úvod	1
2	Informační systémy	2
2.1	Enterprise Resource Planning (ERP)	2
2.2	Advanced Planning and Scheduling (APS)	4
2.3	Integrace informačních systémů	4
2.3.1	Úrovně integrace	4
3	Původní integrační rozhraní v AIMTEC a.s.	7
3.1	Úroveň integrace	7
3.2	Kontext systému	7
3.3	Funkce rozhraní	8
3.3.1	Tabulky systému APS	8
3.3.2	Transformace datových struktur z ERP do APS	9
3.3.3	Transformace datových struktur z APS do ERP	10
3.4	Realizace rozhraní	10
3.4.1	Realizace transformace dat z ERP do APS	11
4	Specifikace požadavků zadavatele	15
4.1	Obecný popis	15
4.1.1	Rozsah projektu	15
4.1.2	Kontext systému	16
4.1.3	Provozní prostředí	18
4.1.4	Omezení návrhu a implementace	18
4.1.5	Uživatelská dokumentace	19
4.1.6	Předpoklady a závislosti	19
4.2	Funkce rozšířeného rozhraní	19
4.2.1	Kontrola konzistence dat z pohledu APS	19
4.2.2	Archivace dat z importních tabulek	21
4.2.3	Archivace dat z chybových tabulek	22
4.2.4	Archivace dat z finálních tabulek	23
4.2.5	Logování rozhraní	23
4.2.6	Zasílání e-mailových notifikací	24
4.3	Požadavky na vnější rozhraní	25
4.3.1	Uživatelské rozhraní	25
4.3.2	Hardwarové rozhraní	25

4.3.3	Softwarová rozhraní	25
4.3.4	Komunikační rozhraní	26
4.4	Další mimofunkční požadavky	26
4.4.1	Výkonnostní požadavky	26
4.4.2	Bezpečnostní požadavky	26
4.4.3	Kvalitativní parametry	26
5	Návrh struktur rozšiřujících funkcí	27
5.1	Kontrola konzistence dat z pohledu APS	27
5.1.1	Chyby	27
5.1.2	Importní tabulky	28
5.1.3	Chybové tabulky	29
5.1.4	Kontrolní procedury	30
5.1.5	Běh rozhraní	31
5.2	Archivace dat z importních tabulek	33
5.2.1	Archivní importní tabulky	33
5.2.2	Záznamy k archivaci	33
5.2.3	Údržba archivních importních tabulek	34
5.3	Archivace dat z chybových tabulek	34
5.3.1	Archivní chybové tabulky	34
5.3.2	Záznamy k archivaci	35
5.3.3	Údržba archivních chybových tabulek	36
5.4	Archivace dat z finálních tabulek	36
5.4.1	Archivní finální tabulky	36
5.4.2	Archivační procedury	37
5.4.3	Údržba archivních finálních tabulek	37
5.5	Logování rozhraní	38
5.5.1	Záznamy logu	38
5.5.2	Logovací tabulka	39
5.5.3	Logovací procedury	39
5.6	Zasílání e-mailových notifikací	41
5.6.1	Atribut <code>Is_Notified</code>	41
5.6.2	Tabulka e-mailových notifikací	41
5.6.3	Notifikační procedury	42
6	Implementace rozšiřujících funkcí	44
6.1	Rozdělení zdrojových souborů	44
6.2	Použité technologie a komponenty	45
6.2.1	Microsoft SQL Server	45
6.2.2	Jazyk T-SQL	46

6.2.3	SQL Server Agent	46
6.2.4	Database Mail	47
6.2.5	Ostatní technologie	48
7	Testování integračního rozhraní	49
7.1	Jednotkové testy	49
7.2	Testování integrace komponent	50
7.3	Nasazení testovací verze	50
8	Závěr	52
	Seznam zkratk	53
	Literatura	55
	Seznam příloh	57
A	Tabulky původního rozhraní	59
A.1	Názvy tabulek – transformace z ERP do APS	59
A.2	Struktura tabulky Item	60
B	Procedury původního rozhraní	61
C	Běh původního rozhraní	62
D	Tabulky rozšířeného rozhraní	63
D.1	Názvy tabulek – transformace z ERP do APS	63
D.2	Společné atributy importních tabulek	64
D.3	Atributy chybové tabulky pro položky	64
D.4	Atributy logovací tabulky	65
D.5	Atributy tabulky e-mailových notifikací	65
E	Procedury rozšířeného rozhraní	66
E.1	Názvy procedur – transformační a kontrolní procedury	66
E.2	Hierarchie notifikačních procedur	67
F	Běh rozšířeného rozhraní	68
G	Kontroly pro položky	70
H	Záznamy logu pro položky	71

I	Uživatelská dokumentace	73
I.1	Vytvoření instalátoru	73
I.2	Instalace databázové aplikace	74
I.2.1	Vytvoření základní databáze	75
I.2.2	Instalace databázových skriptů	77
I.3	Konfigurace databázové aplikace	78
I.3.1	Nastavení globálních proměnných	78
I.3.2	Nastavení SQL jobů	78
I.3.3	Nastavení e-mailových notifikací	79
I.4	Spuštění databázové aplikace	81
I.5	Vstupy a výstupy databázové aplikace	82
J	Obsah CD	84
J.1	Struktura obsahu CD	84
J.2	Popis obsahu CD	84

1 Úvod

Efektivní plánování výroby je v dnešní době klíčem k úspěchu každé výrobní společnosti. Při správném naplánování lze docílit efektivního využití zdrojů, vysoké přesnosti dodacích termínů, optimálního stavu zásob atd., což je nezbytné, pokud chce společnost uspět v konkurenčním prostředí.

Důležitou součástí výrobní společnosti je *Enterprise Resource Planning (ERP)* systém, řešení pro plánování podnikových zdrojů. Systémy této kategorie se zabývají integrací a automatizací velkého množství procesů, které souvisejí s produkčními činnostmi podniku (podrobněji popsáno v kapitole 2.1). Součástí těchto systémů je mimo jiné i plánování výroby. Výrazně vyšší efektivitu lze však dosáhnout se současným použitím *Advanced Planning and Scheduling (APS)* systému, řešením pro pokročilé plánování a rozvrhování. Tyto systémy slouží jako nadstavba ERP systémů a zabývají se výrobním plánováním až po úroveň detailního dílenského rozvrhování (podrobněji v kapitole 2.2).

Právě těmito systémy se zabývá společnost AIMTEC a.s., konkrétně implementací, provozem a servisem APS. Klíčovou částí je integrace obou systémů, k tomu používá společnost AIMTEC a.s. integrační rozhraní, které oba systémy integruje na datové vrstvě. Stávající rozhraní však již v současné době není dostačující.

Cílem této práce je rozšíření stávajícího rozhraní pro plánovací systém vyvíjeného ve společnosti AIMTEC a.s., které zajišťuje přenos dat mezi systémy ERP a APS. V rámci rozšíření dochází k přidání dalších funkcionalit: kontroly konzistence dat z pohledu APS systému, archivace dat, zasílání e-mailových notifikací a logování rozhraní.

V práci nejprve popisují obecnou problematiku informačních systémů a jejich integrace, dále pak systémy ERP a APS.

V další části práce popisují stávající integrační rozhraní mezi APS systémem firmy AIMTEC a.s. a ERP systémem zákazníka. Definují nové požadavky na integrační rozhraní včetně jejich detailní specifikace, návrh rozšiřujících funkcí integračního rozhraní a v implementační části popisují technologie a komponenty použité při implementaci těchto nových funkcí integračního rozhraní.

Na závěr se zabývám testováním rozšířeného integračního rozhraní, posouzením dosažených výsledků a celkovým zhodnocením práce.

2 Informační systémy

„*Informační systém (IS)* je soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování a uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení.“ [8]

Tato kapitola se zabývá informačními systémy v podniku – *Podnikovými informačními systémy* – dále jen *informační systémy*.

V literatuře se můžeme setkat s celou řadou různorodých definic, pro tuto práci však považuji za nejvhodnější výše uvedenou. Díky tomu existuje několik rovin chápání tohoto pojmu.

Tři hlavní roviny jsou:

- **informační systém podporovaný ICT** – informace zpracované a zapsané nejčastěji prostřednictvím relační databáze,
- **informační systém formalizovaný** – informace uložené na dalších (mnohdy „klasických“) nosičích – dokladech, formulářích, zprávách, ale i podporované aplikacemi ICT,
- **obecně komplexní socioekonomický informační systém** – informace, které nejsou dosud zaznamenány v databázi, jiné elektronické podobě a ani nejsou na žádném formuláři (např. zkušenosti lidí) [1].

Z pohledu této práce je nejdůležitější první rovina chápání – tou se budu zabývat. Mezi nejvýznamnější a nejvyužívanější komplexní informační systémy patří systémy kategorie *ERP*.

2.1 Enterprise Resource Planning (ERP)

Enterprise Resource Planning, neboli *Plánování podnikových zdrojů*, je kategorie podnikových informačních systémů. Tyto systémy integrují a automatizují velké množství procesů, které souvisejí s produkčními činnostmi podniku. Jedná se zejména o výrobu, logistiku, distribuci, správu majetku, prodej, fakturaci, účetnictví a údržbu [3].

Stejně jako u pojmu *informační systémy*, i zde existuje mnoho definic, které definují pojem *ERP* z různých úhlů pohledu.

Za ERP jsou považovány:

- aplikace, které představují softwarová řešení užívaná k řízení podnikových dat a pomáhající k plánování celého logistického řetězce od nákupu přes sklady po výdej materiálu, řízení obchodních zakázek od jejich přijetí až po expedici, včetně plánování vlastní výroby a s tím spojené finanční a nákladové účetnictví i řízení lidských zdrojů,
- hotový software, který podniku umožňuje automatizovat a integrovat jeho hlavní podnikové procesy, sdílet společná data a umožnit jejich dostupnost v reálném čase,
- podniková databáze, do které jsou zapisovány všechny důležité podnikové transakce – v této databázi jsou data zpracována, monitorována a na jejím základě reportována [1].

ERP systémy se skládají z modulů, které pokrývají zejména následující funkční oblasti:

- **logistika** – celá podniková logistika (nákup, skladování, výroba, prodej, plánování zdrojů),
- **finance** – finanční, nákladové a investiční účetnictví a podnikový controlling,
- **personalistika** – lidské zdroje [1].

Novým trendem je integrace podniku a jeho IS s okolím směrem k zákazníkům, dodavatelům a partnerům. Tato potřeba dala vzniknout systémům s označením *ERP II*. Ty se skládají z „klasického“ ERP (výše popsaného) a dalších součástí, které jsou:

- **SCM (Supply Chain Management)** – řízení dodavatelského řetězce,
- **CRM (Customer Relationship Management)** – řízení vztahu se zákazníkem,
- **BI (Business Intelligence)** – manažerský informační systém [1].

2.2 Advanced Planning and Scheduling (APS)

Advanced Planning and Scheduling, neboli *Pokročilé plánování a rozvrhování*, je oblast speciálních aplikací v rámci podnikového IS zabývající se výrobním plánováním až po úroveň detailního dílenského rozvrhování.

„APS charakterizuje současně synchronizované plánování všech zdrojů s respektováním všech známých omezení. V systému se definují výchozí podmínky i vstupní parametry a systém APS má následně za úkol nalézt optimální variantu řešení. Se změnou vstupních parametrů se mohou měnit i výsledná doporučení systému. Optimalizační algoritmy pracují na základě kritériálních funkcí, kdy je každý požadavek ohodnocen.“ [1]

2.3 Integrace informačních systémů

V současné době se zvyšují požadavky na množství funkcionalit podnikových informačních systémů. Nejedná se o zvyšování počtu funkcí jednoho informačního systému, ale o vzájemnou provázanost a spolupráci jednotlivých informačních systémů a aplikací v organizaci. Důležitým aspektem každého systému je tedy možnost jeho integrace do stávajícího prostředí IS v organizaci.

2.3.1 Úrovně integrace

Integrace informačních systémů a aplikací lze provádět na různých úrovních. Za klíčové jsou považovány následující úrovně řešení integrace:

- úroveň dat,
- úroveň aplikačních rozhraní (API¹),
- úroveň uživatelského rozhraní.

Úroveň dat

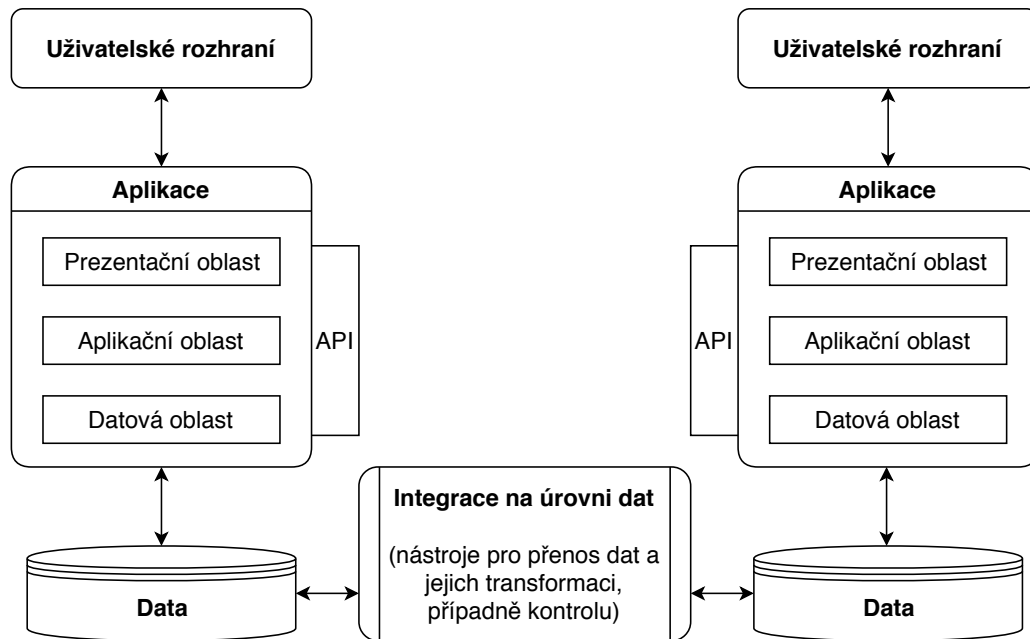
Integrace na úrovni dat spočívá v ignorování existence aplikací. Data se přesouvají přímo mezi datovými zdroji (viz obrázek 2.1) – nejčastěji se jedná o relační databáze. Obvykle se při tom využívá *datových pump (ETL²)*.

¹Application Programming Interface (Aplikační programové rozhraní)

²Extract, Transformation, Load (Extrakce, transformace a nahrání dat)

Datové pumpy zahrnují následující kroky:

1. extrakce dat ze zdrojového prostředí,
2. transformace dat dle požadavků cílového prostředí,
3. vložení dat do cílového prostředí [5].



Obrázek 2.1: Integrace na úrovni dat, Zdroj: Vlastní zpracování dle [5, 6]

Úroveň aplikačních rozhraní (API)

Předpokladem použití integrace na úrovni aplikačních rozhraní je existence definovaného aplikačního programového rozhraní (API). Tato úroveň integrace se využívá zejména u komplexního řešení integrace více aplikací. Používají se zde dva přístupy:

- tvůrce aplikace připravil sadu konkrétních rozhraní ke svému aplikačnímu balíku – funkce těchto rozhraní jsou využívány jinými aplikacemi při komunikaci s aplikačním balíkem,
- tvůrce integrační platformy (speciálních integračních nástrojů) připravil sadu rozhraní k neznámějším aplikačním balíkům.

Může se stát, že rozhraní připravená tvůrcem aplikačního balíku nevyhovují požadavkům volající aplikace, proto většinou existuje kombinace obou

výše zmíněných přístupů. Rozhraní připravená tvůrcem jsou poté doplněna o adaptér, který transformuje data do struktur, které jsou kompatibilní s volající aplikací. Prostřednictvím této integrace je možno přistupovat k obchodní logice aplikace (aplikační oblasti) i datové oblasti a prostřednictvím ní k datům [5].

Úroveň uživatelského rozhraní

Při integraci na úrovni uživatelského rozhraní dochází k nahrazení koncového uživatele obsluhujícího aplikaci automatizovaným postupem, který čeká na interakci aplikace a reaguje na ní simulací uživatelského vstupu.

Tento postup se používá velmi zřídka, je vhodný pouze ve specifických situacích. Pokud je aplikace velmi stará, není k dispozici zdrojový kód, dodavatel již zanikl, jde často o jediný možný způsob, jak aplikaci integrovat [6].

3 Původní integrační rozhraní v AIMTEC a.s.

Jednou z hlavních činností společnosti AIMTEC a.s. je implementace systému APS organizacím, které mají již zavedený ERP systém. Z tohoto důvodu zde vzniká potřeba rozhraní, které integruje mezi sebou oba zmíněné systémy.

Takové rozhraní společnost AIMTEC a.s. již vlastní, to je ovšem nedostatečné a je nutné jej rozšířit o další potřebné funkcionality. Stávající rozhraní má pouze dvě základní funkcionality: transformuje datové struktury z ERP systému do struktur APS systému a naopak. Funkce rozhraní jsou detailně popsány v kapitole 3.3.

3.1 Úroveň integrace

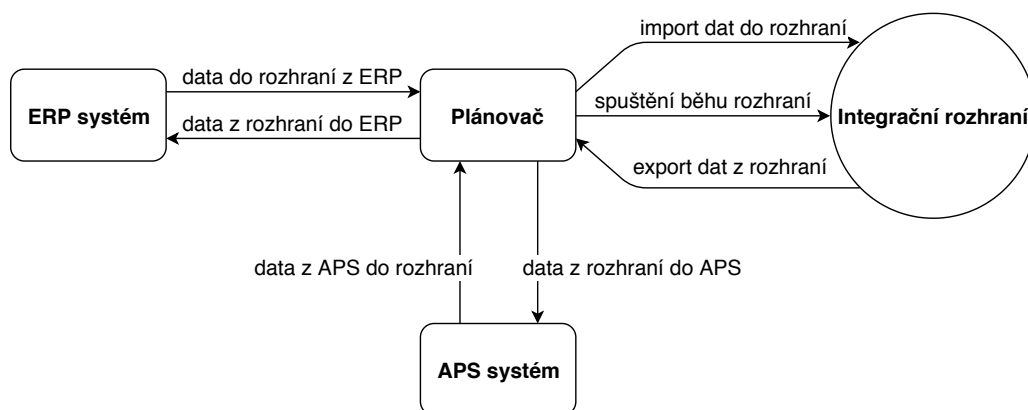
Rozhraní integruje systémy ERP a APS na úrovni dat (viz kapitola 2.3.1) – konkrétně jsou data uložena v relační databázi. Důvodem je skutečnost, že používaný APS systém nemá aplikační programové rozhraní (API).

Integrační rozhraní pracuje pouze s vlastní relační databází. Import dat do databáze ani export dat z databáze integračního rozhraní není součástí stávajícího rozhraní. V důsledku toho musí být data určená k transformaci vložena do relační databáze integračního rozhraní externí entitou (*Plánovačem* – viz kapitola 3.2). Výsledná data jsou po transformaci uložena pouze v databázi rozhraní.

3.2 Kontext systému

Kontext integračního rozhraní je znázorněn na obrázku 3.1. Externí entity integračního rozhraní jsou:

- ERP systém,
- APS systém,
- Plánovač – uživatel v této roli spouští běh rozhraní, vkládá data do databáze integračního rozhraní, exportuje data z rozhraní.



Obrázek 3.1: Kontextový diagram stávajícího integračního rozhraní, Zdroj: Vlastní zpracování

3.3 Funkce rozhraní

Stávající integrační rozhraní má dvě funkcionality:

1. transformuje datové struktury z ERP systému do struktur APS systému (operace *import*),
2. transformuje datové struktury z APS systému do struktur ERP systému (operace *export*).

Transformace datových struktur je široký pojem. Jelikož rozhraní pracuje s relační databází, jedná se konkrétně o transformaci dat z *tabulek A* do *tabulek B*.

3.3.1 Tabulky systému APS

Výrobní data jsou v systému APS uložena v databázi, kde jsou rozřazena do tabulek podle významu. Jsou zde tabulky pro:

- Položky (Items) – název `Item`,
- Zdroje (Resources) – název `Resource`,
- Operace (Operations) – název `Operation`,
- Zakázky (Orders) – název `Order`,
- Zákazníci (Customers) – název `Customer`,
- Směny (Shifts) – název `Shift`,

- Kalendáře (Calendars) – název **Calendar**,
- Kusovník (Bill of materials) s postupy – název **Integrated Master**.

Tabulka **Integrated Master** obsahuje jak kusovník (seznam dílů včetně množství, ze kterých se vyrábí nadřazená položka či finální produkt), tak i jednotlivé postupy pro výrobu. Konkrétně se jedná o: *vstupy (včetně množství)*, *výstupy (včetně množství)*, *výrobní postupy* a *pomocné výrobní prostředky*.

3.3.2 Transformace datových struktur z ERP do APS

Při transformaci datových struktur z ERP do APS používá integrační rozhraní dvě tabulky *pro každou z výše uvedených tabulek* (kapitola 3.3.1) – *importní tabulku* a *finální tabulku*. Například pro *tabulku položek* je použita *importní tabulka položek* a *finální tabulka položek*. Vyjímkou je tabulka kusovníku s postupy – ta je v rámci integračního rozhraní rozdělena na: *tabulku vstupů*, *tabulku výstupů*, *tabulku výrobních postupů* a *tabulku pomocných výrobních prostředků*. Ke každé z těchto tabulek je definována jedna importní tabulka a jedna finální tabulka.

Importní tabulka

Do *importní tabulky* jsou vložena data z ERP systému *Plánovačem*. Struktura *importní tabulky* odpovídá struktuře *tabulky* v ERP systému.

Finální tabulka

Ve *finální tabulce* se nachází data ve strukturách pro APS systém.

Postup transformace

Do *importních tabulek* rozhraní musí být vložena data z ERP systému. Vložení těchto dat zajišťuje *Plánovač*. Ten odpovídá za vložení dat ve správných strukturách. Pokud by data neměla správnou strukturu, rozhraní nebude fungovat správně.

Plánovač následně spustí rozhraní. Po skončení běhu rozhraní jsou data ve *finálních tabulkách* (kompatibilní s APS). *Plánovač* tato data naimportuje do APS.

3.3.3 Transformace datových struktur z APS do ERP

Současné integrační rozhraní umožňuje transformaci z APS do ERP pouze pro tabulky *zakázek* a *operací*. Integrační rozhraní v tomto případě používá dvě tabulky *pro každou z předchozích dvou zmíněných* – *exportní tabulku* a *exportní systémovou tabulku*. Například pro *tabulku zakázek* je použita *exportní tabulka zakázek* a *exportní systémová tabulka zakázek*.

Exportní tabulka

Do *exportní tabulky* jsou vložena data z APS systému *Plánovačem*. Struktura *exportní tabulky* odpovídá struktuře *tabulky* v APS systému.

Exportní systémová tabulka

V *exportní systémové tabulce* se nachází data ve strukturách pro konkrétní ERP systém.

Postup transformace

Obdobně jako u transformace ze struktur ERP do APS, i zde musí do *exportních tabulek* vložít data z APS systému *Plánovač*, který odpovídá za jejich správnou strukturu.

Opět je rozhraní spuštěno *Plánovačem*. Po skončení běhu rozhraní jsou data v *exportních systémových tabulkách* (kompatibilní s ERP). *Plánovač* data naimportuje do konkrétního ERP systému.

3.4 Realizace rozhraní

V této kapitole je popsána realizace původního integračního rozhraní. Jelikož se *rozšíření* integračního rozhraní bude týkat pouze části *transformace datových struktur z ERP do APS*, je v této kapitole uvedena realizace pouze pro tuto část integračního rozhraní.

Rozhraní je vyvinuto v jazyce *T-SQL* pro *Microsoft SQL Server 2016*. Je realizováno jako softwarový balík *skriptů, procedur, funkcí, triggerů, SQL jobů, tabulek a pohledů*, které fungují v prostředí relační databáze. Řešení je dodáváno ve formě instalátoru, ve kterém jsou zabaleny databázové skripty a který umožňuje nainstalovat celý softwarový balík do relační databáze serveru, na němž bude provozováno integrační rozhraní.

Přestože se jedná o prostředí relační databáze, v rozhraní jsou realizovány vazby mezi tabulkami jen velmi zřídka¹ a databáze neodpovídá třetí normální formě. Nejedná se o standardní využití databáze, kde je doporučené dodržovat normální formy, ale pouze o převodní můstek mezi dvěma systémy. Tyto systémy mají vlastní databáze, které plní třetí normální formu a data v nich jsou strukturována obvyklým způsobem. Databáze, která reprezentuje integrační rozhraní, slouží pouze jako soubor tabulek a pohledů a pro manipulaci s daty slouží *procedury*, *funkce*, *joby* a *triggery*. V případě integračního rozhraní ani není reálné, aby databáze splňovala normální formy, jelikož systémy ERP a APS mají odlišné struktury ukládání dat a zákazník by nebyl schopen importovat data do integračního rozhraní tak, aby normální formy splňovala.

3.4.1 Realizace transformace dat z ERP do APS

Tabulky integračního rozhraní

Nyní je nutné uvést přesné názvy tabulek použité v programovém kódu integračního rozhraní. Rozdělení tabulek je popsáno v kapitole 3.3.2.

Tabulky v integračním rozhraní dělíme na:

- finální tabulky – většinou odpovídají názvům tabulek z APS systému (viz kapitola 3.3.1), kromě tabulky kusovníku s postupem (popsáno dále),
- importní tabulky – obsahují stejný název jako odpovídající finální tabulky a příponu `_Import`.

Příkladem mohou být *položky*: finální tabulka – `Item`, importní tabulka – `Item_Import`.

Tabulka *kusovníku s postupem* systému APS (viz kapitola 3.3.1) je rozdělena v rámci integračního rozhraní na čtyři části (viz kapitola 3.3.2):

- *tabulka vstupů*: importní tabulka – `IM_In_Import`, finální tabulka – `IM_In`,
- *tabulka výstupů*: importní tabulka – `IM_OUT_Import`, finální tabulka – `IM_OUT`,

¹Z tohoto důvodu v této práci neuvádím ER model této databáze – obsahoval by minimum vazeb mezi tabulkami.

- *tabulka výrobních postupů*: importní tabulka – `IM_Use_M_Import`, finální tabulka – `IM_Use_M`,
- *tabulka pomocných výrobních prostředků*: importní tabulka – `IM_Use_Sx_Import`, finální tabulka – `IM_Use_Sx`.

Použité názvy tabulek v integračním rozhraní zobrazuje tabulka v příloze A.1.

Z finálních tabulek se data neimportují rovnou do APS systému (to je rozdíl oproti zjednodušenému popisu v kapitole 3.3.2), ale z finálních tabulek se vytvoří pohledy nad daty s přesnými názvy atributů pro APS systém. Z těchto pohledů se poté importují data do APS systému.

Tabulky jsou vytvořeny v databázi pomocí *SQL skriptů*. Název skriptu vždy obsahuje předponu `create.table`, např. tabulku `Item` vytváří skript `create.table.Item`, tabulku `Item_Import` vytváří skript `create.table.Item_Import` atd. Tyto skripty jsou při instalaci integračního rozhraní spuštěné jako první a mají přesně definované pořadí jejich spuštění – jde o tzv. *číslované skripty*. Po vytvoření tabulek jsou nad všemi finálními tabulkami vytvořeny pohledy. Název pohledu je vytvořen z názvu odpovídající finální tabulky a přípony `_View` – např. nad finální tabulkou `Item` je vytvořen pohled `Item_View`.

Každá z tabulek obsahuje jednoznačný identifikátor záznamu (primární klíč) – `ID`. Ostatní atributy má každá z tabulek specifické. V *importních tabulkách* je ovšem pět atributů, které obsahuje každá z importních tabulek. Pomineme-li `ID`, jedná se o:

- `Import_Date` – datum, kdy byl záznam importován do *importních tabulek* rozhraní – vyplňuje *Plánovač* při importu,
- `Import_State` – importní stav, ve kterém se záznam právě nachází, při importu do *importních tabulek* je nastaven *Plánovačem* na `W` – viz dále,
- `Transfer_Date` – datum, kdy byl záznam přenesen z *importní tabulky* do *finální tabulky*,
- `Source_System` – konkrétní systém, ze kterého jsou záznamy importovány.

Struktura pro tabulky *položek* (tabulky `Item_Import`, `Item`) je znázorněna v příloze A.2. Struktura ostatních tabulek je obdobná.

Importní stavy záznamů (**Import_State**)

- *W (waiting)* – v tomto stavu musí být každý záznam ihned po importu do rozhraní – nastavuje *Plánovač*,
- *F (finished)* – tento stav je přiřazen záznamu, který byl úspěšně přenesen do finální tabulky.

T-SQL procedury integračního rozhraní

Do importních tabulek vloží *Plánovač* data určená k transformaci. Ta je prováděna prostřednictvím *uložených T-SQL procedur*. Pro každou *importní tabulkou* jsou vytvořeny uložené procedury, které zajišťují transformaci dat *dané tabulky*, tzn. transformace dat je provedena pro *každou tabulku zvlášť*.

Data jsou transformována do odpovídající *finální tabulky*, např. data z tabulky **Item_Import** se transformují do tabulky **Item**. Poté *Plánovač* importuje data do systému APS z pohledu nad odpovídající *finální tabulkou*, např. pro finální tabulku **Item** je odpovídající pohled **Item_View**.

Uložené procedury a jejich funkcionality:

- hlavní procedura (přípona **_Main**):
 - prostřednictvím této procedury se spouští transformace dat v konkrétní *importní tabulce*,
 - procedura nejprve odstraní záznamy s importním stavem *F* – jedná se o staré záznamy z minulého běhu rozhraní,
 - poté spustí *proceduru pro přenos*,
- procedura pro přenos (přípona **_Transfer**):
 - procedura zajišťuje přenos dat z importní do finální tabulky – přenáší záznamy s importním stavem *W*,
 - procedura nejprve vyplní **Transfe_Date** záznamům určeným k přenosu (importní stav *W*),
 - poté tyto záznamy transformuje z importní tabulky do finální tabulky,
 - následně nastaví úspěšně přeneseným záznamům importní stav *F* v importní tabulce,

- procedura pro vytvoření pohledu (předpona `Create_` a přípona `_View`):
 - procedura vytvoří *pohled* (View) nad daty finální tabulky pro systém APS.

Seznam procedur v integračním rozhraní zobrazuje tabulka v příloze B.

Běh rozhraní

Transformace dat probíhá podle následujících bodů:

1. *Plánovač* odstraní všechny záznamy z finálních tabulek,
2. do importních tabulek vloží *Plánovač* data z ERP systému s importním stavem *W* a datem importu (předchozí záznamy neodstraňuje),
3. následně spustí postupně hlavní procedury všech tabulek,
4. záznamy s importním stavem *F* jsou odstraněny,
5. data jsou transformována a přenesena do finálních tabulek,
6. z finálních tabulek jsou vytvořeny pohledy pro APS systém,
7. z těchto pohledů *Plánovač* importuje data do APS.

Běh rozhraní pro *položky* (tabulky `Item_Import`, `Item`) je zobrazen prostřednictvím diagramu aktivit v příloze C. Pro ostatní tabulky je běh obdobný.

4 Specifikace požadavků zadavatele

Současný stav integračního rozhraní není dostačující. Rozhraní musí být doplněno o další funkcionality. V této kapitole definuji požadavky zadavatele (firma AIMTEC a.s.) na rozšíření stávajícího integračního rozhraní (viz kapitola 3).

4.1 Obecný popis

4.1.1 Rozsah projektu

Projekt se zabývá rozšířením stávajícího integračního rozhraní mezi ERP a APS systémy v AIMTEC a.s. Rozsah projektu odpovídá v součtu časové dotaci, která je stanovena pro *Projekt 5 a Bakalářskou práci*, tj. přibližně 460 hodinám.

Stávající funkce rozhraní (viz kapitola 3.3):

- transformace datových struktur z ERP systému do struktur APS systému,
- transformace datových struktur z APS systému do struktur ERP systému.

Rozšíření stávajícího integračního rozhraní se bude týkat pouze té části rozhraní, která zajišťuje transformaci datových struktur z ERP systému do struktur APS systému. V rámci tohoto rozšíření dojde k přidání následujících funkcionalit:

- **kontrola konzistence dat z pohledu APS systému:**
 - zda záznam obsahuje příslušné atributy, které jsou potřebné pro naplánování výroby systémem APS,
 - zda jsou atributy ve správném formátu,
 - zda jsou data úplná (např. pokud se v kusovníku pracuje s určitou položkou, musí být tato položka uvedena v tabulce položek),
 - k přenosu záznamu do finální tabulky nedojde, pokud obsahuje fatální chybu,

- vytvoření zprávy o konkrétních chybách v konkrétních záznamech v podobě chybové tabulky,
- **archivace dat z importních, chybových a finálních tabulek:**
 - vytvoření archivních tabulek pro data z importních, chybových a finálních tabulek,
 - údržba archivních dat – automatické mazání starých dat,
- **logování rozhraní:**
 - vytvoření zprávy o běhu rozhraní,
 - vytvoření zprávy o spuštění a dokončení běhu procedur – úspěšném či neúspěšném,
- **zasílání e-mailových notifikací:**
 - informování příslušných osob o dokončení běhu rozhraní prostřednictvím e-mailových notifikací,
 - shrnutí chyb v datech importovaných ze systému ERP,

Podrobný popis nových funkcí se nachází v kapitole 4.2.

Přínosy rozšíření integračního rozhraní:

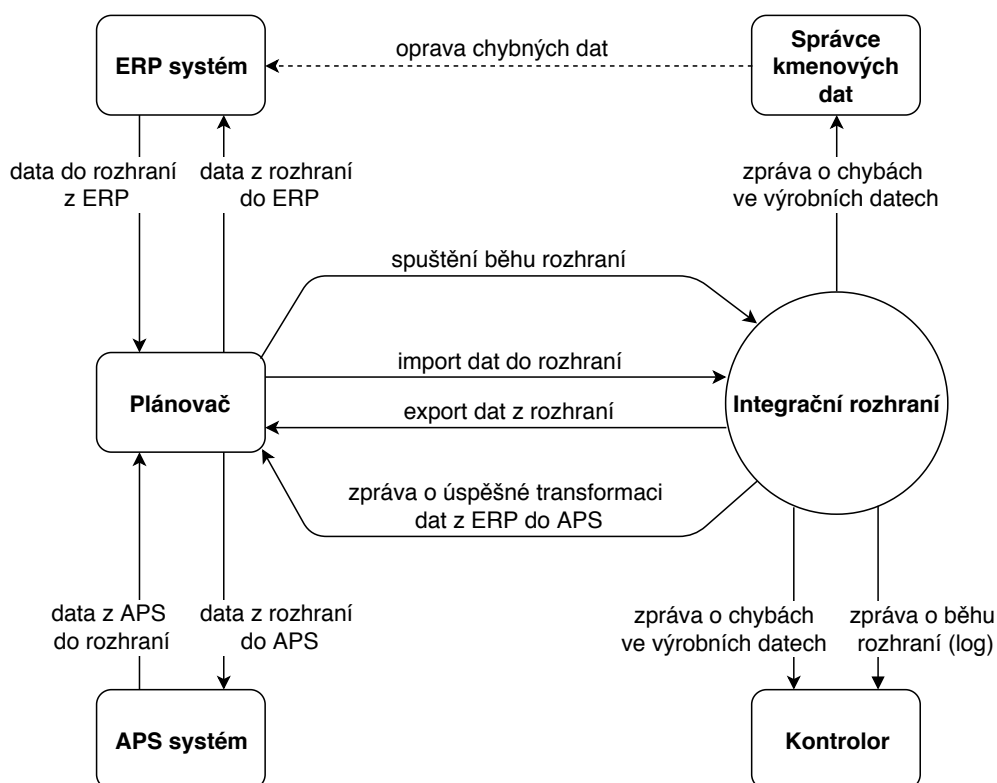
- data budou konzistentní z pohledu APS systému,
- původní i zkontrolovaná data budou uchovávána pro další potřeby,
- diagnostikované chyby budou odeslány příslušným osobám,
- jednotlivé kroky běhu rozhraní budou zaznamenávány pro případnou kontrolu a dohledatelnost chyb.

4.1.2 Kontext systému

Kontext rozšířeného integračního rozhraní je znázorněn na obrázku 4.1. Oproti původnímu rozhraní zde přibyly dvě externí entity: *Správce kmenových dat* a *Kontrolor*.

Externí entity rozšířeného integračního rozhraní tedy jsou:

- ERP systém,
- APS systém,
- Plánovač,
- Správce kmenových dat,
- Kontrolor.



Obrázek 4.1: Kontextový diagram rozšířeného integračního rozhraní, Zdroj: Vlastní zpracování

Plánovač

Uživatel v této roli spouští běh rozhraní, vkládá data do databáze integračního rozhraní, exportuje data z rozhraní. Oproti původnímu rozhraní ještě dostává informaci o tom, zda byla data z ERP systému úspěšně transformována do struktur APS systému a zda může data ze struktur pro APS systém naimportovat do APS systému a úspěšně zaplánovat.

Správce kmenových dat

Uživatel v této roli dostává informace o konkrétních chybách ve výrobních datech, které byly zjištěny v rámci kontroly konzistence dat z pohledu APS při transformaci struktur. Na rozdíl od *Plánovače* může opravovat chyby ve výrobních datech.

Kontrolor

Uživatel v této roli přijímá nejen informace o konkrétních chybách ve výrobních datech, ale také dostává informace (log) o běhu integračního rozhraní.

4.1.3 Provozní prostředí

Softwarový produkt je určen pro společnosti, které mají již implementovaný ERP systém. Tento produkt vyžaduje server a klientský počítač (je možná instalace na jednom zařízení s využitím lokálního databázového serveru). Software využívá systém řízení báze dat (SŘBD) Microsoft SQL Server 2016, který je umístěn na serveru. Tento SŘBD je spravován z klientského počítače prostřednictvím Microsoft SQL Server 2016 Management Studio.

Minimální požadavky na server:

- operační systém: Windows 2012 R2 a vyšší,
- procesor: 64 bitová architektura, frekvence minimálně 2 GHz,
- pevný disk: 128 GB a více,
- paměť RAM: 4 GB a více,
- databázový server: Microsoft SQL Server 2016 Standard Edition SP1 a vyšší.

Minimální požadavky na klientský počítač:

- operační systém: Windows 7 a vyšší,
- ostatní software: Microsoft SQL Server Management Studio 2016 a vyšší.

4.1.4 Omezení návrhu a implementace

Pro implementaci integračního rozhraní je použit jazyk T-SQL, Microsoft SQL Server Management Studio 2016 a SŘBD Microsoft SQL Server 2016 Standard Edition SP1. Implementací se detailně zabývá kapitola 6. Omezením je také struktura dat. Ta musí být striktně dodržena zákazníkem.

4.1.5 Uživatelská dokumentace

K tomuto softwarovému produktu bude dodávána uživatelská dokumentace v písemné podobě, která bude popisovat nastavení a spuštění systému pro bezchybnou funkčnost. Dále bude popisovat vstupy a výstupy integračního rozhraní.

4.1.6 Předpoklady a závislosti

Softwarový produkt je závislý na datech dodávaných zákazníkem. Data musí být striktně ve stanovených strukturách, v opačném případě software nebude pracovat správně.

4.2 Funkce rozšířeného rozhraní

V následující kapitole jsou popsány nové funkce, které mají být zapracovány do původního rozhraní – tím vznikne rozšířené integrační rozhraní. Pokud není uvedeno jinak, struktura dat zůstává nezměněna oproti původnímu rozhraní.

4.2.1 Kontrola konzistence dat z pohledu APS

Nejprve je nutné rozšířit *importní stavy záznamů* oproti původnímu rozhraní.

Importní stavy záznamů

V kapitole 3.4.1 byly zmíněny *importní stavy záznamů*. K těmto záznamům budou vytvořeny další tři typy: *C*, *WARNING* a *ERROR*. *Importní stavy položek* tedy jsou:

- *W (waiting)* – v tomto stavu musí být každý záznam ihned po importu do rozhraní – nastavuje *Plánovač*,
- *C (checked)* – tento stav má záznam poté, co byl zkontrolován a neobsahuje žádnou chybu (*ERROR*, *WARNING*),
- *WARNING* – tento stav je nastaven záznamu, u kterého byly zjištěny problémy při kontrolách, ale tyto problémy nejsou vážné; záznam **bude** přenesen do finální tabulky,
- *ERROR* – tento stav je nastaven záznamu, u kterého byly zjištěny vážné problémy při kontrolách; záznam **nebude** přenesen do finální tabulky,

- *F (finished)* – tento stav je přiřazen záznamu, který byl přenesen do finální tabulky.

Jako *chyby* jsou označovány záznamy, které mají importní stav *ERROR* nebo *WARNING*.

Popis

Tato funkce bude kontrolovat data importovaná z ERP systému, zda jsou konzistentní z pohledu systému APS. Bude kontrolovat:

- zda záznam obsahuje příslušné atributy, které jsou potřebné pro naplánování výroby systémem APS,
- zda jsou atributy ve správném formátu,
- zda jsou data úplná (např. pokud se v kusovníku pracuje s určitou položkou, musí být tato položka uvedena v tabulce položek).

Na základě kontroly dat budou vyhodnoceny *importní stavy záznamů* a bude vytvořena tabulka konkrétních chyb v konkrétních záznamech. Na konci vyhodnocovací funkce přiřadí každému záznamu z importních tabulek jeden z importních stavů: *C*, *WARNING* nebo *ERROR*. Záznamům se stavem *WARNING* a *ERROR* přiřadí ještě příčinu chyby (nebo příčiny chyb – může jich být i více) a chybový kód (nebo chybové kódy). Následně budou do *finální tabulky* přeneseny pouze záznamy se stavy *C* nebo *WARNING*, kterým bude následně v importní tabulce přiřazen stav *F*.

Události a odpovědi

Funkce bude spuštěna při každém přenosu dat z ERP systému do APS systému. Každý nově importovaný záznam bude mít nastavený importní stav na *W*.

Tato funkce každý nový záznam zkontroluje a případné chyby uloží do *chybové tabulky*. Dále vyhodnotí záznamy v chybové tabulce a podle toho každému záznamu z importní tabulky přiřadí jeden ze stavů: *C*, *WARNING* nebo *ERROR*.

Funkční požadavky

1. Funkce přiřadí každému záznamu z importních tabulek, který má stav *W*, datum kontroly, pokud dosud není vyplněno.

2. Na záznamech s aktuálním datem kontroly a s importním stavem *W* provede postupně každou kontrolu.
3. Chyby uloží do chybové tabulky.
4. Na základě konkrétních chyb vyhodnotí záznamy v importní tabulce a každému z nich přiřadí jeden ze stavů: *C*, *WARNING* nebo *ERROR*.
5. Záznamům se stavy *WARNING* a *ERROR* přiřadí příčinu chyby (popř. příčiny chyb) a chybový kód (popř. chybové kódy).
6. Transformuje pouze záznamy se stavy *C* a *WARNING* do finální tabulky.

4.2.2 Archivace dat z importních tabulek

Popis

Funkce vytvoří archivní importní tabulky pro data z importních tabulek. Archivovat se budou:

- data, která byla při minulém běhu rozhraní úspěšně přenesena do finálních tabulek (jsou ve stavu *F* a nemají příčinu chyby ani chybový kód),
- data, která byla notifikována (viz kapitola 4.2.6),
- data, která jsou starší než předem zvolený počet dní.

Funkce bude zajišťovat údržbu archivních importních tabulek, především odstraňování starých záznamů.

Události a odpovědi

Pokud má záznam v importní tabulce nastaven importní stav na hodnotu *F* a zároveň nemá nastavenou příčinu chyby, dojde k archivaci záznamu. Pokud je datum importu (`Import_Date`) starší než zadaný počet dní, dojde k archivaci záznamu. Pokud byl záznam notifikován, také dojde k jeho archivaci. V ostatních případech není se záznamem manipulováno.

Funkční požadavky

1. Funkce zarchivuje záznamy, které mají importní stav *F* a není nastavena příčina chyby ani chybový kód.

2. Funkce zarchivuje záznamy, které mají datum importu starší než zadaný počet dní.
3. Funkce zarchivuje záznamy, které byly notifikovány.
4. Funkce bude spuštěna při každém přenosu dat z *importních tabulek* do *finálních tabulek*.
5. Funkce bude zajišťovat odstraňování starých záznamů v archivních importních tabulkách.

4.2.3 Archivace dat z chybových tabulek

Popis

Funkce vytvoří archivní chybové tabulky pro data z chybových tabulek. Archivovat se budou:

- záznamy chybových tabulek, které odpovídají záznamům v importních tabulkách, jejichž datum importu je starší než předem zvolený počet dní,
- záznamy chybových tabulek, které odpovídají záznamům v importních tabulkách, jež byly notifikovány.

Dále bude funkce zajišťovat údržbu archivních chybových tabulek, především odstraňování starých záznamů.

Události a odpovědi

Pokud je datum importu (`Import_Date`) záznamu v importní tabulce starší než zadaný počet dní, dojde k archivaci odpovídajících chybových záznamů (pokud existují). Pokud byl záznam v importní tabulce notifikován, dojde k archivaci odpovídajících chybových záznamů (pokud existují). V ostatních případech není se záznamy chybové tabulky manipulováno.

Funkční požadavky

1. Funkce zarchivuje záznamy chybových tabulek, které odpovídají záznamům v importních tabulkách, jejichž datum importu je starší než předem zvolený počet dní.
2. Funkce zarchivuje záznamy chybových tabulek, které odpovídají záznamům v importních tabulkách, jež byly notifikovány.

3. Funkce bude spuštěna při každém přenosu dat z *importních tabulek* do *finálních tabulek*.
4. Funkce bude zajišťovat odstraňování starých záznamů v archivních chybových tabulkách.

4.2.4 Archivace dat z finálních tabulek

Popis

Funkce vytvoří archivní finální tabulky pro data z finálních tabulek. Archivace bude spouštěna periodicky. Archivovat se budou všechny záznamy z finálních tabulek.

Dále se bude funkce starat o údržbu archivních finálních tabulek – mazání starých dat (starší než zadaný počet dní). Data budou odstraňována periodicky.

Události a odpovědi

Funkce bude spouštěna periodicky. Archivace dat z finálních tabulek a odstraňování starých dat z archivních finálních tabulek bude spouštěno nezávisle, každé bude spouštěno dle vlastní periody.

Funkční požadavky

1. Funkce zarchivuje všechny záznamy z každé finální tabulky.
2. Funkce odstraní záznamy z archivní finální tabulky, které jsou starší než zadaný počet dní.
3. Funkce bude spouštět archivaci a mazání záznamů automaticky dle zadané periody.

4.2.5 Logování rozhraní

Popis

Funkce bude zaznamenávat běh rozhraní a podávat informace o jednotlivých krocích běhu rozhraní, konkrétně o spuštění a dokončení jednotlivých procedur – úspěšném či neúspěšném.

Informace budou uloženy v tabulce logu, každý krok běhu rozhraní bude mít jednoznačný identifikátor – kvůli dohledatelnosti případných chyb v běhu rozhraní.

Události a odpovědi

Funkce bude spouštěna automaticky v každé proceduře určené k logování.

Funkční požadavky

1. Funkce bude zaznamenávat běh rozhraní a podávat informace o jednotlivých krocích běhu.
2. Funkce bude podávat informaci, zda daná procedura byla spuštěna a zda její běh skončil úspěchem či neúspěchem.
3. Výstupem bude tabulka se záznamy o běhu rozhraní.
4. Funkce bude spouštěna automaticky v každé proceduře určené k logování.

4.2.6 Zasílání e-mailových notifikací

Popis

Tato funkce bude informovat o dokončení běhu rozhraní prostřednictvím e-mailových notifikací. Budou se odesílat tři typy e-mailů:

1. *e-mail s chybami* – bude obsahovat absolutní četnost chybových (*ERROR*, *WARNING*) a korektních záznamů a tabulku s konkrétními chybami v konkrétních záznamech (viz kapitola 4.2.1),
2. *e-mail s logem* – bude obsahovat logovací tabulku integračního rozhraní,
3. *e-mail s výsledkem* – bude obsahovat pouze informaci o tom, zda transformace dat proběhla úspěšně, tzn. zda výsledek *kontroly konzistence dat z pohledu APS* skončil bez záznamů s importním stavem *ERROR*.

V závislosti na konkrétní roli uživatele jsou odeslány příslušné e-maily. *Kontrolorovi* jsou odeslány všechny výše zmíněné e-maily, tzn. e-mail s chybami, e-mail s logem a e-mail s výsledkem. *Správci kmenových dat* je odeslán e-mail s chybami a e-mail s výsledkem. *Plánovači* je odeslán pouze e-mail s výsledkem.

Události a odpovědi

Funkce bude spuštěna po každém přenosu dat z importních tabulek do finálních tabulek, pokud bude nastavený příznak notifikací. Při každém spuštění funkce se odešlou příslušné e-maily *Kontrolorovi*, *Správci kmenových dat* a *Plánovači* a chybovým záznamům (*ERROR*, *WARNING*) se nastaví příznak, že byly notifikovány.

Funkční požadavky

1. Funkce odešle *Kontrolorovi* e-mail s chybami, e-mail s logem a e-mail s výsledkem.
2. Funkce odešle *Správci kmenových dat* e-mail s chybami a e-mail s výsledkem.
3. Funkce odešle *Plánovači* e-mail s výsledkem.
4. Funkce nakonec nastaví chybovým záznamům (*ERROR*, *WARNING*) příznak, že byly notifikovány.

4.3 Požadavky na vnější rozhraní

4.3.1 Uživatelské rozhraní

Softwarový produkt neobsahuje uživatelské rozhraní.

4.3.2 Hardwarové rozhraní

Softwarový produkt neobsahuje hardwarové rozhraní.

4.3.3 Softwarová rozhraní

Pro tento produkt jsou softwarovými rozhraními systémy ERP a APS. Jedná se o systémy, jejichž integrací se vyvíjený softwarový produkt zabývá. Systém ERP je v tomto případě obecným řešením plánování podnikových zdrojů na straně zákazníka, nezáleží na konkrétním produktu a dodavateli tohoto systému. Systém APS je v tomto případě konkrétním řešením pokročilého plánování a rozvrhování, který implementuje společnost AIMTEC a.s.

4.3.4 Komunikační rozhraní

Vyvíjený software využívá SŘBD Microsoft SQL Server 2016, se kterým komunikuje prostřednictvím Microsoft SQL Server 2016 Management Studio. Pro spuštění jobů se využívá SQL Server Agent (součást SQL Serveru). Pro odesílání e-mailových notifikací je využita služba Database Mail (též součástí SQL Serveru), která využívá SMTP server.

4.4 Další mimofunkční požadavky

4.4.1 Výkonnostní požadavky

Software bude nasazen zejména u výrobních firem, které mají velké množství dat. Proto je maximální deklarovaná doba běhu rozhraní 2 hodiny. Předpokládá se denní frekvence spouštění rozhraní, vždy před začátkem směny v organizaci.

4.4.2 Bezpečnostní požadavky

Každý uživatel bude autentifikován pomocí uživatelského jména a hesla.

4.4.3 Kvalitativní parametry

Kód softwarového produktu musí být napsán dle firemní kultury psaní kódu společnosti AIMTEC a.s.

5 Návrh struktur rozšiřujících funkcí

Rozšířené integrační rozhraní je postaveno na již výše zmíněném původním integračním rozhraní ve společnosti AIMTEC a.s. (viz kapitola 3). Rozšíření spočívá v doplnění původního rozhraní o rozšiřující funkce, které jsou specifikovány v kapitole 4.2. Vychází se z původních struktur, které jsou upraveny a rozšířeny.

V následující kapitole popisují podrobný návrh struktur rozšiřujících funkcí integračního rozhraní.

5.1 Kontrola konzistence dat z pohledu APS

Kontrola konzistence dat z pohledu APS tvoří spolu s *transformací datových struktur z APS do ERP* hlavní funkcionalitu integračního rozhraní. Tato funkce kontroluje, zda data importovaná z ERP systému jsou konzistentní z pohledu systému APS. Jsou vytvořeny tabulky chyb, které podávají informace o konkrétních chybách v konkrétních záznamech. Na základě těchto tabulek jsou vyhodnoceny importní stavy záznamů, které jsou uloženy v importních tabulkách. Podrobná specifikace této funkce je uvedena v kapitole 4.2.1.

V původním rozhraní byly definovány *pro každou z tabulek* kapitoly 3.3.1 dvě tabulky – *importní tabulka* a *finální tabulka* (viz příloha A.1). Nyní je nutné nadefinovat třetí typ tabulky, a tím je *chybová tabulka* (obsahuje příponu `_Import_Error`). Například pro *položky* jsou definovány tabulky `Item_Import` (importní tabulka), `Item_Import_Error` (chybová tabulka) a `Item` (finální tabulka). Použité názvy tabulek v rozšířeném integračním rozhraní zobrazuje tabulka v příloze D.1.

5.1.1 Chyby

Nad každou importní tabulkou jsou definovány kontroly a odpovídající chyby. Každá z chyb má definován:

- stav chybového záznamu (typ chyby):
 - *ERROR* – pokud se jedná o vážnou chybu, nebo
 - *WARNING* – pokud se jedná o chybu, která není vážná,

- kód chyby:
 - skládá se ze tří písmen a ze tří číslic,
 - první písmeno: E – pokud se jedná o *ERROR*, W – pokud se jedná o *WARNING*,
 - druhé a třetí písmeno: identifikátor tabulky (např. pro tabulku `Item_Import` je to IT),
 - číslice: identifikátor chyby pro danou tabulku,
- popis příčiny chyby:
 - stručný popis konkrétní chyby, popř. nápověda na její opravu.

Například pokud existuje v importní tabulce položek (`Item_Import`) záznam, který nemá vyplněný atribut `Code`, v chybové tabulce bude pro daný záznam chyba se stavem: *ERROR*, kódem chyby: EIT001 a popisem příčiny chyby: *Item code is missing. Fill in the field "Code".;*

Konkrétní výčet definovaných chyb pro *položky* je obsažen v tabulce přílohy G. Pro ostatní tabulky jsou chyby podobné.

5.1.2 Importní tabulky

Importní tabulky rozšířeného integračního rozhraní vycházejí z importních tabulek původního rozhraní, ale jsou doplněny o další důležité atributy (počet ani názvy tabulek se oproti původnímu rozhraní nemění). V rámci *kontroly konzistence dat z pohledu APS* se jedná o atributy:

- `Import_Error_Code` – kódové označení chyby nebo chyb (pokud je jich více, jsou odděleny středníkem) – v případě bezchybného záznamu je hodnota *null*,
- `Import_Error_Reason` – popis příčiny chyby nebo chyb (pokud je jich více, jsou odděleny středníkem) – v případě bezchybného záznamu je hodnota *null*,
- `Check_Date` – datum, kdy byl záznam zkontrolován.

Společné atributy všech importních tabulek jsou podrobněji popsány v příloze D.2¹.

¹Význam některých atributů uvedených v příloze D.2 bude vysvětlen dále v textu.

5.1.3 Chybové tabulky

Chybová tabulka obsahuje výčet konkrétních chyb pro konkrétní záznam dané importní tabulky. Záznam importní tabulky může obsahovat i více chyb, díky tomu je mezi danou importní tabulkou a chybovou tabulkou vazba 1:N (jeden záznam importní tabulky může mít více odpovídajících záznamů v chybové tabulce). Každá z chyb má definováno, zda se jedná o *WARNING* nebo *ERROR*, a tato informace je zanesena do chybové tabulky (konkrétně do atributu *Import_State* – viz dále). Každá z chybových tabulek obsahuje následující atributy:

- *ID* – jednoznačný identifikátor záznamu (primární klíč),
- *<název tabulky>_Import_ID* – jednoznačný identifikátor záznamu v importní tabulce, jedná se o cizí klíč, pomocí něhož je realizována vazba 1:N,
- *<název tabulky>_Import_Import_Date* – datum, kdy byl odpovídající záznam v importní tabulce importován,
- *Import_State* – stav chybového záznamu (typ chyby – *WARNING* nebo *ERROR*),
- *Import_Error_Code* – kódové označení dané chyby (v tomto případě nedochází ke zřetězování chybových kódů – jednomu záznamu odpovídá jeden chybový kód),
- *Import_Error_Reason* – popis příčiny dané chyby (v tomto případě nedochází ke zřetězování popisů příčin chyby – jednomu záznamu odpovídá jeden popis příčiny chyby).

Atributy chybové tabulky pro *položky* jsou podrobněji popsány v příloze D.3. Pro ostatní tabulky jsou atributy obdobné – liší se pouze v prefixu některých atributů.

Na základě záznamů chybové tabulky je rozhodnuto o importním stavu záznamu v importní tabulce. Vyhodnocení importních stavů pro záznamy v importní tabulce je následovné:

- pokud záznamu z importní tabulky přísluší v chybové tabulce alespoň jeden záznam se stavem *ERROR*, záznamu v importní tabulce bude přiřazen importní stav *ERROR*,
- pokud záznamu z importní tabulky přísluší v chybové tabulce alespoň jeden záznam se stavem *WARNING* a žádný se stavem *ERROR*, záznamu v importní tabulce bude přiřazen importní stav *WARNING*,

- pokud záznamu z importní tabulky nepřísluší žádný záznam z chybové tabulky, záznamu v importní tabulce bude přiřazen importní stav *C* – záznam neobsahuje žádnou chybu.

Dále je všem záznamům importní tabulky, které mají odpovídající záznam či záznamy v chybové tabulce, přiřazeno kódové označení chyby či chyb a popis příčiny chyby či chyb. Pokud má daný záznam v importní tabulce více chyb, chybové kódy i popisy příčin chyb se řetězí pomocí středníku.

5.1.4 Kontrolní procedury

V kapitole 3.4.1 jsou uvedeny základní uložené procedury původního integračního rozhraní. V rámci *kontroly konzistence dat z pohledu APS* je nutné k těmto procedurám definovat další procedury, které budou kontrolovat jednotlivé záznamy a vyhodnocovat případné chyby. Pro každou z tabulek jsou tyto procedury dvě: *kontrolní procedura* a *vyhodnocovací procedura*.

Procedury pro kontrolu a vyhodnocování jsou:

- kontrolní procedura (přípona `_Check`):
 - kontroluje jednotlivé záznamy importní tabulky se stavem *W* a aktuálním datem kontroly podle předem definovaných kontrol,
 - jednotlivé chyby pro konkrétní záznamy importní tabulky uloží do chybové tabulky s odpovídajícím stavem chybového záznamu (*ERROR* nebo *WARNING*), kódovým označením chyby a popisem příčiny chyby,
 - nakonec spustí vyhodnocovací proceduru,
- vyhodnocovací procedura (přípona `_Check_Evaluation_State`):
 - vyhodnocuje stavy záznamů importní tabulky na základě příslušných chyb v chybové tabulce,
 - záznamům v importní tabulce přiřadí jeden ze stavů: *C*, *WARNING* nebo *ERROR* (postup vyhodnocení importních stavů pro záznamy v importní tabulce je popsáno v kapitole 5.1.3),
 - dále přiřadí chybným záznamům v importní tabulce kódové označení chyby či chyb a popis příčiny chyby či chyb (pro bezchybné záznamy (stav *C*) jsou tyto hodnoty nastaveny na *null*).

Pro tabulky *zdrojů* je ještě navíc vytvořena procedura `Resource_Import_Common_Check` a pro tabulky *zakázek* jsou navíc vytvořeny procedury `Order_Import_Common_Check` a `Order_Import_Manufacturing_Order_Check`. Tyto procedury jsou vytvořeny z důvodu přehlednosti kontrol a snadnější rozšiřitelnosti. Jsou obaleny příslušnými *kontrolními procedurami*, tudíž jejich existenci můžeme při dalším popisu zanedbat.

Po vytvoření *kontrolních procedur* a *vyhodnocovacích procedur* je nutné upravit *proceduru pro přenos*, která je popsána v kapitole 3.4.1. *Procedura pro přenos*:

- zajišťuje přenos dat z importní do finální tabulky – přenáší záznamy s importním stavem *C* a *WARNING*,
- nejprve vyplní `Transfer_Date` záznamům určeným k přenosu (importní stav *C* a *WARNING*),
- poté tyto záznamy transformuje z importní tabulky do finální tabulky,
- následně nastaví úspěšně přeneseným záznamům importní stav *F* v importní tabulce.

Seznam transformačních a kontrolních procedur rozšířeného integračního rozhraní zobrazuje tabulka v příloze E.1. *Pohledy* nad daty finální tabulky pro systém APS zůstávají nezměněny oproti původnímu integračnímu rozhraní.

5.1.5 Běh rozhraní

Transformace dat původního rozhraní je popsána v kapitole 3.4.1. V této kapitole je popsán běh rozhraní, který se týká transformace dat z ERP do APS včetně kontroly konzistence dat z pohledu APS systému. Jedná se o hlavní funkcionalitu integračního rozhraní. Běh probíhá podle následujících bodů:

1. *Plánovač* odstraní všechny záznamy z finálních tabulek,
2. do importních tabulek vloží *Plánovač* data z ERP systému s importním stavem *W* a datem importu (předchozí záznamy neodstraňuje),
3. následně spustí sekvenčně všechny hlavní procedury (přípona `_Main`) – pořadí spuštění je definováno níže,
4. jsou *archivovány záznamy chybových tabulek* (detailně v kapitole 5.3),

5. záznamy, které byly archivovány, jsou z chybových tabulek odstraněny,
6. jsou *archivovány záznamy importních tabulek* (detailně v kapitole 5.2),
7. záznamy, které byly archivovány, jsou z importních tabulek odstraněny,
8. následně jsou spuštěny kontrolní procedury, které provedou kontroly nad záznamy importních tabulek a případné chyby uloží do chybových tabulek,
9. poté jsou spuštěny vyhodnocovací procedury, které na základě chyb z chybových tabulek přiřadí záznamům v importních tabulkách jeden ze stavů: *C*, *WARNING* nebo *ERROR*,
10. záznamy v importních tabulkách se stavy *C* a *WARNING* jsou transformovány a přeneseny do finálních tabulek prostřednictvím procedur pro přenos,
11. importní stavy přenesených záznamů do finálních tabulek jsou nastaveny na *F*,
12. jsou odeslány *e-mailové notifikace*, pokud jsou nastaveny (detailně v kapitole 5.6),
13. z pohledů nad finálními tabulkami importuje *Plánovač* data do APS.

Na konci běhu rozhraní jsou tedy data v importních tabulkách buď se stavem *F*, nebo se stavem *ERROR*. Běh rozhraní pro *položky* (tabulky *Item_Import* a *Item*) je zobrazen prostřednictvím diagramu aktivit v příloze F. Pro ostatní tabulky je běh obdobný.

Jak již bylo zmíněno výše, spuštění běhu rozhraní je realizováno sekvencím spuštěním všech hlavních procedur. Některé definované kontroly porovnávají hodnoty atributů mezi více tabulkami, je tedy nutné dodržet pořadí spuštění hlavních procedur:

1. *Item_Import_Main*
2. *Resource_Import_Main*
3. *Customer_Import_Main*
4. *IM_Use_M_Import_Main*
5. *IM_OUT_Import_Main*
6. *IM_In_Import_Main*
7. *IM_Use_Sx_Import_Main*

8. `Order_Import_Main`
9. `Operation_Import_Main`
10. `Shift_Import_Main`
11. `Calendar_Import_Main`

5.2 Archivace dat z importních tabulek

Archivace dat z importních tabulek je další funkcí rozšířeného integračního rozhraní. Tato funkce se stará o přenos záznamů z importních tabulek do archivních importních tabulek na základě definovaných kritérií (viz níže). Podrobná specifikace této funkce je uvedena v kapitole 4.2.2.

5.2.1 Archivní importní tabulky

Pro potřeby archivace dat z importních tabulek je nutné nadefinovat další typ tabulky pro každou *importní tabulku*, a tím je *archivní importní tabulka* (obsahuje příponu `_Import_Archive`). Například pro importní tabulku `Item_Import` je definována archivní importní tabulka `Item_Import_Archive`.

Struktura archivních importních tabulek je shodná se strukturou importních tabulek, kromě atributu ID, který je v tomto případě také primárním klíčem, ale není automaticky generovaný. Důvodem je skutečnost, že ID záznamu importní tabulky je také přenášeno do archivní importní tabulky, tzn. ID záznamu v archivní importní tabulce odpovídá ID příslušného záznamu v importní tabulce.

5.2.2 Záznamy k archivaci

K archivaci jsou určeny záznamy importní tabulky, které:

- byly při minulém běhu rozhraní přeneseny do finální tabulky a neobsahovaly chybu – tzn. záznamy s `Import_State = F` a `Import_Error_Reason = null`, nebo
- obsahovaly chybu (*ERROR* či *WARNING*) a byly notifikovány – tzn. záznamy s `Is_Notified = 1` (detailně v kapitole 5.6.1), nebo
- mají datum importu starší než předem zvolený počet dní – tzn. `Import_Date < actualDate - archiveTimeOffset`.

Proměnná *actualDate* udává aktuální datum, proměnná *archiveTimeOffset* udává dobu, po kterou bude záznam uchován v importní nebo chybové tabulce (pokud je záznam v importní nebo chybové tabulce déle než tato doba, je archivován), a je možno ji zvolit dle potřeby.

Archivace dat z importních tabulek je spouštěna v hlavních procedurách (přípona *_Main*) na jejich začátcích. Z tohoto důvodu dojde k přenesení záznamů do archivních importních tabulek po jejich importu až při opětovném spuštění rozhraní.

Záznamy přenesené do *archivních importních tabulek* jsou z *importních tabulek* odstraněny.

5.2.3 Údržba archivních importních tabulek

Součástí funkce pro *archivaci dat z importních tabulek* je také *údržba archivních importních tabulek*. V tomto případě je údržbou myšleno *periodické odstraňování starých záznamů*.

Odstraňování starých záznamů zajišťuje procedura *CUST_Archive_Tables_Maintenance*. Ta na základě data importu záznamu vyhodnocuje, zda záznam není starší než předem zvolený počet dní – tzn. $Import_Date < actualDate - commonPeriod$. Proměnná *actualDate* udává aktuální datum, proměnná *commonPeriod* udává dobu, po kterou bude záznam v archivní importní tabulce (pokud je záznam v archivní importní tabulce déle než tato doba, je odstraněn), a je možno ji zvolit dle potřeby.

Aby bylo docíleno periodického odstraňování starých záznamů, musí být procedura *CUST_Archive_Tables_Maintenance* spouštěna periodicky prostřednictvím SQL jobu.

5.3 Archivace dat z chybových tabulek

Další rozšiřující funkcí integračního rozhraní je *Archivace dat z chybových tabulek*. Tato funkce přenáší záznamy z chybových tabulek do archivních chybových tabulek na základě definovaných kritérií (viz níže). Detailní specifikace této funkce je uvedena v kapitole 4.2.3.

5.3.1 Archivní chybové tabulky

Obdobně jako u *archivace dat z importních tabulek*, i zde je nutné nejprve nadefinovat další typ tabulky. Pro každou *chybovou tabulku* je definována

archivní chybová tabulka (obsahuje příponu `_Import_Error_Archive`). Například pro chybovou tabulku `Item_Import_Error` je definována archivní chybová tabulka `Item_Import_Error_Archive`.

Struktura archivních chybových tabulek je opět shodná se strukturou chybových tabulek, kromě atributu `ID`, který je v tomto případě také primárním klíčem, ale není automaticky generovaný. Je to kvůli tomu, že `ID` záznamu chybové tabulky je také přenášeno do archivní chybové tabulky, tzn. `ID` záznamu v archivní chybové tabulce odpovídá `ID` příslušného záznamu v chybové tabulce.

5.3.2 Záznamy k archivaci

Archivovat se budou záznamy chybové tabulky, které:

- odpovídají chybovým záznamům (*ERROR* či *WARNING*) v importní tabulce, jež byly notifikovány – tzn. odpovídající záznam v importní tabulce má `Is_Notified = 1` (detailně v kapitole 5.6.1), nebo
- odpovídají záznamům v importní tabulce, jejichž datum importu je starší než předem zvolený počet dní – tzn. odpovídající záznam v importní tabulce má `Import_Date < actualDate - archiveTimeOffset`.

Proměnná `actualDate` opět udává aktuální datum, proměnná `archiveTimeOffset` opět udává dobu, po kterou bude záznam uchován v importní nebo chybové tabulce, a je možno ji zvolit dle potřeby.

Lze tedy konstatovat, že záznamy importní tabulky, které byly notifikovány či mají staré datum importu, budou archivovány a k nim odpovídající záznamy z chybové tabulky budou také archivovány. V případě záznamů importní tabulky, které byly bez chyb přeneseny do finální tabulky, dojde též k jejich archivaci, tyto záznamy však nemají odpovídající záznamy v chybové tabulce (budou tedy archivovány pouze záznamy importní tabulky).

Archivace dat z chybových tabulek je spouštěna v hlavních procedurách (přípona `_Main`) na jejich začátcích před archivací dat z importních tabulek. Z tohoto důvodu dojde k přenesení záznamů do archivních chybových tabulek po jejich importu až při opětovném spuštění rozhraní.

Podobně jako u archivace dat z importních tabulek, i zde jsou záznamy přenesené do *archivních chybových tabulek* odstraněny z *chybových tabulek*.

5.3.3 Údržba archivních chybových tabulek

Obdobně jako u *archivace dat z importních tabulek*, i v tomto případě je nutné *periodické odstraňování starých záznamů*, tentokrát z archivních chybových tabulek.

Stejně jako u předchozí funkce, i zde se o odstraňování starých záznamů stará procedura `CUST_Archive_Tables_Maintenance`. Ta vyhodnotí, zda odpovídající záznam v importní tabulce je starší než předem zvolený počet dní (stejně jako v kapitole 5.2.3), a pokud ano, odstraní odpovídající záznamy v chybové tabulce.

Jak již bylo zmíněno v kapitole 5.2.3, k docílení periodického odstraňování starých záznamů je procedura spouštěna periodicky prostřednictvím SQL jobu.

5.4 Archivace dat z finálních tabulek

Třetí funkcí pro archivaci dat je *Archivace dat z finálních tabulek*. Ta přenáší záznamy z finálních tabulek do archivních finálních tabulek. Podrobná specifikace této funkce je uvedena v kapitole 4.2.4.

5.4.1 Archivní finální tabulky

Pro účely této funkce je nezbytné nadefinovat další typ tabulky. Pro každou *finální tabulku* je definována *archivní finální tabulka* (obsahuje příponu `_Archive`). Například pro finální tabulku `Item` je definována archivní finální tabulka `Item_Archive`.

Struktura archivních finálních tabulek je shodná se strukturou finálních tabulek, kromě:

- atributu `ID` – v případě archivní finální tabulky se jedná také o primární klíč (stejně jako ve finální tabulce), ale není generovaný automaticky, `ID` záznamu finální tabulky je přenášeno do archivní finální tabulky, tzn. `ID` záznamu v archivní finální tabulce odpovídá `ID` příslušného záznamu ve finální tabulce,
- atributu `Archive_Date` – tento atribut obsahuje pouze archivní finální tabulka, jedná se o datum archivace daného záznamu, atribut je povinný, typ atributu: `datetime`.

5.4.2 Archivační procedury

Přenos záznamů do archivních finálních tabulek se liší od předchozích dvou archivací. V přechozích dvou případech byla archivace dat součástí hlavních procedur, v tomto případě se o archivaci starají *archivační procedury*. Pro každou *finální tabulku* je definována *archivační procedura* (obsahuje předponu `Archive_`). Například pro finální tabulku `Item` je definována archivační procedura `Archive_Item`.

Archivační procedura po jejím spuštění zkopíruje všechny záznamy z finální tabulky do archivní finální tabulky a vyplní hodnotu atributu `Archive_Date` aktuálním datem. Záznamy z finální tabulky neodstraňuje, o to se stará *Plánovač* před každým spuštěním integračního rozhraní.

K docílení periodičnosti archivace záznamů jsou všechny archivační procedury spouštěny periodicky prostřednictvím SQL jobu. Doporučená perioda spouštění procedur je jeden den.

5.4.3 Údržba archivních finálních tabulek

Součástí této funkce je také *údržba archivních finálních tabulek*. Jedná se o *periodické odstraňování starých záznamů*.

O odstraňování starých záznamů se starají dvě procedury: `Table_Maintenance` a `Archive_Table_Maintenance`.

Procedura `Table_Maintenance` je generická a má dva povinné parametry: *tableName* a *archiveDeleteOffset*. Parametr *tableName* udává konkrétní název archivní finální tabulky, parametr *archiveDeleteOffset* udává dobu, po kterou bude záznam uchován v archivní finální tabulce. Procedura po spuštění odstraní záznamy tabulky, jejíž název je předán parametrem *tableName*, a jejichž datum archivace je starší než zadaný počet dní předaný parametrem *archiveDeleteOffset*, tzn. záznamy, které splňují podmínku $Archive_Date < actualDate - archiveDeleteOffset$, kde *actualDate* je aktuální datum.

Procedura `Archive_Table_Maintenance` obaluje proceduru `Table_Maintenance`, kterou spouští nad každou archivní finální tabulkou (prostřednictvím parametru *tableName*) se zvolenou dobou uchování záznamu v archivní finální tabulce (prostřednictvím parametru *tableName*).

K docílení periodického odstraňování starých záznamů je procedura `Archive_Table_Maintenance` spouštěna periodicky prostřednictvím SQL jobu.

5.5 Logování rozhraní

Další rozšiřující funkcí integračního rozhraní je *Logování rozhraní*. Tato funkce zaznamenává jednotlivé kroky běhu rozhraní a výsledky ukládá do *logovací tabulky*. Podrobná specifikace této funkce je uvedena v kapitole 4.2.5.

5.5.1 Záznamy logu

Log běhu integračního rozhraní je reprezentován *logovací tabulkou* (viz dále) a skládá se z jednotlivých záznamů (stejně jako logovací tabulka). Záznamy logu jsou rozděleny do sedmi kategorií podle toho, o jaké části integračního rozhraní podávají informaci. Každý záznam také obsahuje *zprávu a komentář*.

Kategorie záznamů logu (seřazeny v pořadí, jak jdou za sebou při logování běhu rozhraní):

- Kategorie 0 – zaznamenává smazání záznamů z finálních tabulek.
- Kategorie 1 – zaznamenává vložení dat do importních tabulek.
- Kategorie 2 – zaznamenává začátek a konec archivace dat z importních a chybových tabulek.
- Kategorie 3 – zaznamenává začátek a konec běhu kontrolních a vyhodnocovacích procedur.
- Kategorie 4 – zaznamenává začátek a konec běhu procedur pro přenos.
- Kategorie 5 – zaznamenává jednotlivé kroky při odesílání e-mailových notifikací.
- Kategorie 99 – zaznamenává začátek a konec archivace dat z finálních tabulek.

Každá z výše uvedených kategorií zahrnuje několik kroků, které se liší *zprávou a komentářem* v záznamu logu. Kompletní přehled těchto kroků pro *položky* je uveden v tabulce přílohy H. Pro ostatní tabulky jsou kroky obdobné.

5.5.2 Logovací tabulka

Dále je nutné nadefinovat *logovací tabulku*, do které se budou ukládat jednotlivé záznamy logu běhu rozhraní. Logovací tabulka nese název `Import_Event_Log`. Do této tabulky jsou ukládány záznamy běhu rozšířeného integračního rozhraní pro všechny tabulky.

Logovací tabulka obsahuje pět atributů:

- `ID` – jednoznačný identifikátor záznamu (primární klíč),
- `Log_Number` – číslo kategorie záznamu v logu,
- `Table` – název tabulky, ke které daný záznam v logu náleží,
- `Time` – datum vytvoření záznamu v logu,
- `Message` – zpráva k danému záznamu v logu,
- `Comment` – komentář k danému záznamu v logu.

Atributy logovací tabulky jsou podrobněji popsány v tabulce přílohy D.4.

5.5.3 Logovací procedury

O zaznamenávání jednotlivých kroků rozhraní a ukládání záznamů do logovací tabulky se starají *logovací procedury* – lze je rozdělit do dvou skupin:

1. generická logovací procedura – `Log_Import_Event`,
2. specifické logovací procedury – obsahují předponu `Log_`, název odpovídající finální tabulky a příponu `_Import_Event` (například pro tabulku *položek* je definována specifická logovací procedura `Log_Item_Import_Event`).

Generická logovací procedura `Log_Import_Event` je jen jedna a obsahuje povinné parametry:

- *tableName* – název tabulky, nad kterou je právě prováděno logování,
- *logNumber* – číslo kategorie aktuálního záznamu logu,
- *message* – zpráva k aktuálnímu záznamu v logu,
- *comment* – komentář k aktuálnímu záznamu v logu.

Generická logovací procedura po spuštění vloží záznam do logovací tabulky s využitím předaných parametrů, do atributu `Time` vloží aktuální datum. Tato procedura tedy obsahuje výkonný kód a je vždy volána z některé specifické logovací procedury.

Pro každou tabulku z kapitoly 3.3.1 je definována specifická logovací procedura. Každá z těchto procedur obaluje generickou logovací proceduru a za parametr `tableName` vkládá název konkrétní tabulky, např. pro *položky* je definována specifická logovací procedura `Log_Item_Import_Event`, která volá generickou logovací proceduru `Log_Import_Event` s parametrem `tableName = Item`. Každá ze specifických logovacích procedur obsahuje tři povinné parametry `logNumber`, `message` a `comment`, jejichž významy jsou stejné jako u generické logovací procedury, a tyto parametry jsou pouze předány generické logovací proceduře.

Specifické logovací procedury jsou volány v jednotlivých krocích běhu integračního rozhraní, jež jsou sledovány, a do parametrů `logNumber`, `message` a `comment` jsou vkládány konkrétní hodnoty podle aktuálního kroku. Konkrétní způsob volání specifických logovacích procedur závisí na kategorii sledovaného kroku. Pro kategorii 0 a 1 jsou specifické logovací procedury volány prostřednictvím triggerů. U ostatních kategorií je volání specifických logovacích procedur součástí sledovaných procedur.

Způsob volání specifických logovacích procedur:

- Kategorie 0 – pro každou finální tabulku je vytvořen *after delete trigger* (obsahuje příponu `_A_D`), např. pro finální tabulku `Item` je definován *after delete trigger* `Item_A_D`, význam: po odstranění záznamů z tabulky `Item` dojde ke spuštění specifické logovací procedury,
- Kategorie 1 – pro každou importní tabulku je vytvořen *after insert trigger* (obsahuje příponu `_A_I`), např. pro importní tabulku `Item_Import` je definován *after insert trigger* `Item_Import_A_I`, význam: po vložení záznamů do tabulky `Item_Import` dojde ke spuštění specifické logovací procedury,
- Kategorie 2 – specifické logovací procedury jsou spouštěny v hlavních procedurách,
- Kategorie 3 – specifické logovací procedury jsou spouštěny v kontrolních a vyhodnocovacích procedurách,
- Kategorie 4 – specifické logovací procedury jsou spouštěny v procedurách pro přenos,

- Kategorie 5 – specifické logovací procedury jsou spouštěny v jednotlivých notifikačních procedurách,
- Kategorie 99 – specifické logovací procedury jsou spouštěny v archivačních procedurách.

5.6 Zasílání e-mailových notifikací

Poslední funkcí rozšířeného integračního rozhraní je *Zasílání e-mailových notifikací*. Tato funkce informuje o dokončení běhu rozhraní prostřednictvím e-mailových notifikací. Podle role uživatele (*Kontrolor*, *Správce kmenových dat* a *Plánovač*) odesílá příslušné e-maily. Podrobná specifikace této funkce je uvedena v kapitole 4.2.6.

5.6.1 Atribut `Is_Notified`

V předchozím textu již byl atribut `Is_Notified` zmíněn. O tento atribut jsou v rámci rozšířeného integračního rozhraní doplněny všechny *importní* a *archivní importní tabulky*. Jedná se o příznak, který udává, zda byl chybový záznam (*ERROR* či *WARNING*) v importní tabulce již notifikován. V případě, že záznam ještě notifikován nebyl, hodnota příznaku je 0, v opačném případě je hodnota příznaku 1. Příznak tedy má význam jen pro chybové záznamy, pro korektní záznamy je jeho hodnota 0. Popis atributu je uveden také v tabulce přílohy D.2.

Na základě hodnoty tohoto atributu je rozhodováno o archivaci dat z importních tabulek a z chybových tabulek (viz kapitoly 5.2.2 a 5.3.2). K přeznačování hodnot atributu `Is_Notified` dochází v notifikačních procedurách (viz kapitola 5.6.3).

5.6.2 Tabulka e-mailových notifikací

Pro potřeby e-mailových notifikací je definována *tabulka e-mailových notifikací*, která má název `Email_Notifications`. V tabulce e-mailových notifikací jsou uloženy názvy importních tabulek, které mají být notifikovány, názvy sloupců, které mají být v tabulce *e-mailu s chybami*, adresy odesílateľů a příjemců, jednotlivé role a předmět e-mailu. Záznam v této tabulce reprezentuje konfiguraci e-mailu pro konkrétní importní tabulku a konkrétní roli uživatele.

V tabulce e-mailových notifikací je uložena také tabulka `Import_Event_Log`. U záznamu pro tuto tabulku je jako u jediného

vyplněný atribut `Log_Time_Offset`, který udává časový úsek ve dnech, pro který má být odeslán log integračního rozhraní v rámci *e-mailu s logem*.

Informace z tabulky e-mailových notifikací jsou čteny a využívány notifikačními procedurami.

Atributy tabulky e-mailových notifikací jsou:

- `ID` – jednoznačný identifikátor záznamu (primární klíč),
- `Table_Name` – název tabulky, jedná se buď o *importní tabulku*, která má být notifikována, nebo o *logovací tabulku* `Import_Event_Log`,
- `Notified_Column_Names` – názvy sloupců, která mají být notifikovány (vzájemně odděleny čárkou),
- `To_Address` – adresa příjemce e-mailové notifikace,
- `From_Address` – adresa odesílatele e-mailové notifikace,
- `Subject` – předmět e-mailu (v notifikačních procedurách bývá doplněn o další informace),
- `Comment` – komentář k záznamu,
- `Role` – role náležící danému záznamu (reprezentujícímu konkrétní konfiguraci), *Kontrolor* odpovídá v tabulce názvu `Inspector`, *Správce kmenových dat* odpovídá názvu `Master_Data_Admin` a *Plánovač* názvu `Planner`,
- `Log_Time_Offset` – časový úsek ve dnech, pro který má být odeslán log v rámci *e-mailu s logem*, je vyplněn pouze u záznamu s názvem tabulky `Import_Event_Log`, pro ostatní záznamy je hodnota *null*.

Atributy tabulky e-mailových notifikací jsou podrobněji popsány v tabulce přílohy D.5.

5.6.3 Notifikační procedury

V této podkapitole jsou popsány *notifikační procedury*, které se starají o vytvoření příslušných e-mailů a jejich zasílání. Z důvodu velkého množství těchto procedur je pro přehlednost v příloze E.2 uvedena jejich hierarchie.

Hlavní notifikační procedurou je procedura `Notify_Users`, která je volána z *hlavních procedur* (přípona `_Main`) integračního rozhraní na jejich koncích. Obsahuje dva vstupní parametry: *tableName* (název importní tabulky, které náleží hlavní procedura, ze které je procedura

Notify_Users volána) a *smtpProfileName* (vytvořený e-mailový profil). Procedura Notify_Users spouští postupně procedury Notify_Inspector, Notify_Master_Data_Admin a Notify_Planner a po jejich vykonání přeznačí příznak *Is_Notified* u chybových záznamů v importní tabulce (předané parametrem *tableName*), které měly doposud *Is_Notified* = 0, na *Is_Notified* = 1 (tzn. záznamy byly notifikovány).

Procedura Notify_Inspector zajišťuje odesílání příslušných notifikačních e-mailů *Kontrolorovi*, Notify_Master_Data_Admin zajišťuje odesílání příslušných notifikačních e-mailů *Správci kmenových dat* a Notify_Planner zajišťuje odesílání příslušných notifikačních e-mailů *Plánovači*. Těmto procedurám jsou předány parametry *tableName* a *smtpProfileName* z procedury Notify_Users. Všechny tyto procedury fungují tak, že si nejprve ověří, zda v tabulce e-mailových notifikací (*Email_Notifications*) existuje záznam pro danou tabulku (*tableName*) s příslušnou rolí (*Kontrolor – Inspector*, *Správce kmenových dat – Master_Data_Admin*, *Plánovač – Planner*) a pokud ano, načtou si informace, které záznam obsahuje (příjemce, odesílatel...). Poté sekvenčně volají některé nebo všechny procedury *Notify_Import_Errors*, *Notify_Import_Log* a *Notify_Import_Success* podle toho, o jakou roli se jedná.

Procedura *Notify_Import_Errors* zajišťuje odesílání *e-mailu s chybami*, *Notify_Import_Log* zajišťuje odesílání *e-mailu s logem* a *Notify_Import_Success* zajišťuje odesílání *e-mailu s výsledkem*. Z těchto procedur jsou volány funkce *getEmailBodyError*, *getEmailBodyLog* a *getEmailBodySuccess*, které vytvoří příslušný řetězec reprezentující tělo e-mailu. Procedura *Send_Notification* poté přidá potřebné značky do tohoto řetězce a zavolá proceduru *Send_Mail*, která s příslušnými parametry volá systémovou proceduru pro odeslání e-mailu.

Součástí procedur *Notify_Import_Errors* a *Notify_Import_Success* je procedura *Notify_Import_Counts*, která vypočítává absolutní četnosti chybových a korektních záznamů.

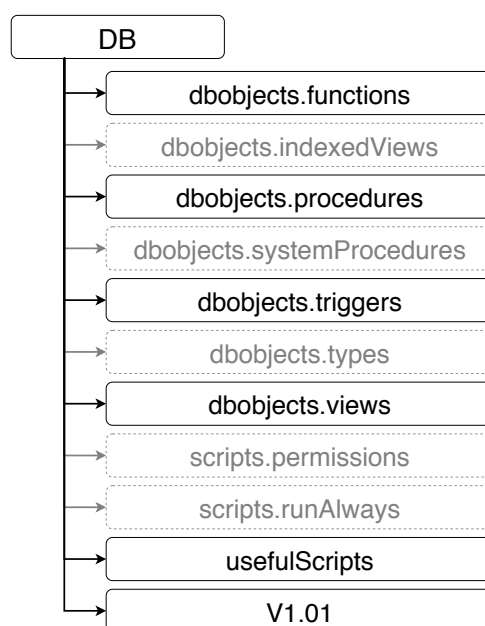
6 Implementace rozšiřujících funkcí

Implementace rozšíření integračního rozhraní proběhla dle specifikace požadavků zadavatele (kapitola 4) a návrhu struktur rozšiřujících funkcí (kapitola 5).

V této kapitole zobrazuji rozdělení zdrojových souborů do adresářů a dále popisuji technologie a komponenty, které byly použity při vývoji tohoto softwaru.

6.1 Rozdělení zdrojových souborů

Rozdělení hlavních zdrojových souborů rozšířeného integračního rozhraní do adresářů ilustruje obrázek 6.1.



Obrázek 6.1: Adresářová struktura hlavních zdrojových souborů, Zdroj: Vlastní zpracování

Význam hlavních adresářů:

- `dbobjects.functions` – obsahuje databázové funkce,
- `dbobjects.procedures` – obsahuje uložené procedury databáze,

- `dbobjects.triggers` – obsahuje databázové triggerly,
- `dbobjects.views` – obsahuje databázové pohledy,
- `usefulScripts` – obsahuje databázové skripty, které usnadňují práci s databází, zejména s jejím počátečním nastavením,
- `V1.01` – obsahuje číslované skripty, tj. skripty, které jsou spouštěny jako první při instalaci databáze a mají přesně určené pořadí spuštění, jedná se zejména o skripty, které zakládají tabulky integračního rozhraní.

6.2 Použité technologie a komponenty

Nejdůležitějšími technologiemi, které byly použity při implementaci rozšíření integračního rozhraní, jsou **Microsoft SQL Server** a jazyk **T-SQL**, který je v produktu Microsoft SQL Server využíván.

Dále byly použity dvě komponenty Microsoft SQL Serveru: **SQL Server Agent** (použit k realizaci SQL jobů) a **Database Mail** (použit k odesílání e-mailových notifikací). Pro účely e-mailových notifikací byl také částečně použit jazyk **HTML** spolu s kaskádovými styly (**CSS**) a **SMTP** server. K testování jednotlivých procedur a funkcí byl použit nástroj **T-SQL Test Tool**.

V této kapitole stručně charakterizují výše zmíněné technologie a uvedu souvislosti použitých technologií s návrhem funkcí.

6.2.1 Microsoft SQL Server

Microsoft SQL server je systém řízení báze dat (SŘBD) pro relační databáze vyvíjený společností Microsoft. Podporuje širokou škálu transakčního zpracování, business intelligence a analytických aplikací v prostředí podnikových informačních systémů.

První verze produktu Microsoft SQL Server byla uvedena na trh v roce 1989 a byla navržena pro operační systém OS/2 (vyvíjen firmou IBM). Na vývoji se kromě firmy Microsoft podílely také firmy Sybase Inc. a Ashton-Tate Corp. Od roku 1994 je SQL Server vyvíjen samostatně firmou Microsoft [9].

Software je nabízen v několika edicích. Nejznámější z nich jsou 4: **Enterprise**, **Standard**, **Developer** a **Express**. Poslední zmíněná edice je jako jediná zdarma, ale obsahuje značná omezení, např. maximální velikost databáze je omezena na 10 GB, edice neobsahuje SQL Server Agent ani Database

Mail. Od verze Microsoft SQL Server 2017 je produkt multiplatformní – je možný provoz na systémech s jádrem Linux [13].

Pro realizaci rozšířeného integračního rozhraní byla použita verze **Microsoft SQL Server 2016 SP1** a edice **Standard** z důvodu nutnosti komponent SQL Server Agent a Database Mail a velikosti databáze, která přesahuje 10 GB.

Pro práci s produktem Microsoft SQL Server jsem použil **Microsoft SQL Server 2016 Management Studio**, které slouží jako grafické uživatelské rozhraní pro SQL Server.

Praktický průvodce všemi funkcemi a možnostmi, které SQL Server nabízí, viz [2].

6.2.2 Jazyk T-SQL

SQL, neboli Structured Query Language (strukturovaný dotazovací jazyk), je standardizovaný strukturovaný dotazovací jazyk, který je používán ke správě relačních databází a vykonávání nejrůznějších operací nad daty uložených v těchto databázích. Příkazy jazyka SQL můžeme rozdělit do tří hlavních kategorií: DDL (data definition language) – příkazy pro definici dat, DML (data manipulation language) – příkazy pro manipulaci s daty a DCL (data control language) – řízení přístupových práv [10].

Jazyk **T-SQL**, neboli Transact-SQL, je sada proprietárních rozšíření standardizovaného jazyka SQL vyvinutých společnostmi Microsoft a Sybase. Sada rozšíření zahrnuje procedurální programování, řízení transakcí, lokální proměnné, funkce pro zpracování řetězců, matematické funkce atd. Tento jazyk je používán v produktu Microsoft SQL Server [11, 15].

6.2.3 SQL Server Agent

SQL Server Agent je komponenta Microsoft SQL Serveru, která umožňuje automaticky vykonávat plánované úkoly v SQL Serveru, které jsou nazývány *joby*. Tato komponenta funguje jako služba Microsoft Windows, a proto může být spuštěna automaticky po startu systému. V rámci jednoho jobu může být nadefinováno více kroků, které mají být vykonány. Každý krok má vlastní úlohu, např. se může jednat o zálohu databáze [14].

V rámci rozšířeného integračního rozhraní používám komponentu SQL Server Agent pro realizaci všech jobů. Jedná se o:

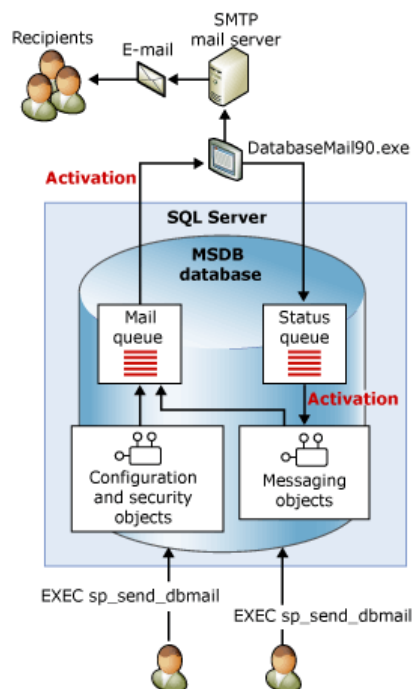
- dosažení periodického odstraňování starých záznamů v archivních importních tabulkách a archivních chybových tabulkách (viz kapitoly 5.2.3 a 5.3.3),

- docílení periodičnosti archivace záznamů do archivní finální tabulky (viz kapitola 5.4.2),
- docílení periodického odstraňování starých záznamů v archivních finálních tabulkách (viz kapitola 5.4.3).

6.2.4 Database Mail

Database Mail je komponenta Microsoft SQL Serveru, která umožňuje odesílat e-maily z SQL Serveru. Prostřednictvím této komponenty mohou databázové aplikace odesílat e-maily uživatelům, jejichž obsahem mohou být například výsledky databázových dotazů. K odesílání e-mailů je využíván protokol SMTP (podrobněji v kapitole 6.2.5) [4].

Pro odeslání e-mailu musí být nejprve volána procedura `sp_send_dbmail`, která vloží zprávu do fronty. Poté je spuštěn externí proces `DatabaseMail.exe`, který zpracuje informace o e-mailu a odešle jej příslušnému e-mailovému SMTP serveru. Ten se pak stará o odeslání e-mailu příjemci. Proces odeslání e-mailu prostřednictvím komponenty Database Mail je ilustrován na obrázku 6.2.



Obrázek 6.2: Proces odeslání e-mailu, Zdroj: [4]

V rámci rozšířeného integračního rozhraní používám **Database Mail** pro realizaci zasílání e-mailových notifikací (viz kapitola 5.6).

6.2.5 Ostatní technologie

HTML a CSS

HTML, neboli Hypertext Markup Language, je standardní značkovací jazyk používaný zejména pro vytváření webových stránek a webových aplikací [17]. CSS, neboli Cascading Style Sheets (kaskádové styly), je jazyk popisující vzhled prvků dokumentu, který je psaný ve značkovacím jazyce [16].

Těchto dvou technologií využívám v rámci rozšířeného integračního rozhraní k vytvoření a naformátování e-mailu zasílaného notifikační funkcí (viz kapitola 5.6).

SMTP

SMTP, neboli Simple Mail Transfer Protocol, je standardní komunikační protokol pro přenos e-mailů na sítích typu TCP/IP. Tento protokol zajišťuje schopnost odesílat a přijímat e-maily [12].

V integračním rozhraní je tento protokol využíván službou Database Mail k odesílání e-mailových notifikací.

T-SQL Test Tool

T-SQL Test Tool je nástroj pro jednoduché psaní a spouštění automatických testů pro programový kód psaný v jazyce T-SQL. Poskytuje API pro psaní *jednotkových testů* [7].

Jednotkové testy jsou druhem testování, jehož cílem je ověření správné funkčnosti jednotek nebo dílčích částí zdrojového kódu [18].

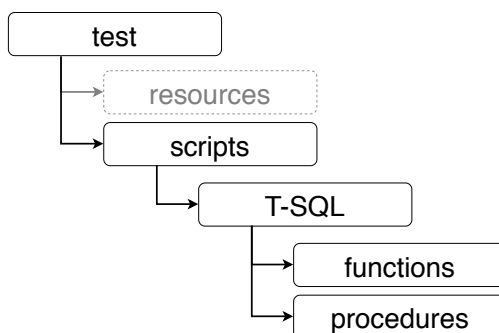
T-SQL Test Tool byl použit v rámci integračního rozhraní k realizaci jednotkových testů, které ověřují správnou funkčnost procedur a funkcí. Toto testování je podrobně popsáno v kapitole 7.1.

7 Testování integračního rozhraní

V této kapitole popisují průběh testování rozšířeného integračního rozhraní. Testování lze rozdělit do tří fází. V první fázi byly testovány jednotlivé procedury a funkce rozšířeného integračního rozhraní prostřednictvím jednotkových testů, ve druhé fázi byla testována integrace všech komponent integračního rozhraní a ve fázi třetí došlo k nasazení testovací verze u zákazníka a ověření funkčnosti integračního rozhraní v reálném prostředí.

7.1 Jednotkové testy

V rámci testování rozšířeného integračního rozhraní byly realizovány jednotkové testy pro jednotlivé procedury a funkce tohoto integračního rozhraní prostřednictvím nástroje T-SQL Test Tool. Tyto testy jsou součástí zdrojových souborů integračního rozhraní, rozdělení jednotkových testů do adresářů ilustruje obrázek 7.1.



Obrázek 7.1: Adresářová struktura jednotkových testů, Zdroj: Vlastní zpracování

Adresář `functions` obsahuje jednotkové testy funkcí, adresář `procedures` obsahuje jednotkové testy procedur. Název každého testu obsahuje předponu `SQLTest_` a dále název procedury či funkce, kterou testuje. Například pro proceduru `Item_Import_Check` je vytvořen test `SQLTest_Item_Import_Check` a je uložen ve složce `procedures`.

Tyto testy fungují na principu porovnávání očekávaných hodnot s reálnými hodnotami. Pokud jsou očekávané hodnoty shodné s reálnými, test je úspěšný, v opačném případě je neúspěšný. Každý z testů běží v transakci,

kde na konci běhu testu je zavolána operace `rollback`, aby vše, co se změnilo v rámci testu, bylo vráceno do původního stavu (tzn. před spuštěním testu).

Jednotkovými testy jsou testovány téměř všechny procedury a funkce integračního rozhraní. Vyjímkou jsou pouze notifikační funkce, které jsou testovány pomocí testovacích dat ručně.

Testování jednotkovými testy bylo finálně úspěšné, všechny jednotkové testy vyšly s pozitivním výsledkem.

7.2 Testování integrace komponent

Po otestování integračního rozhraní jednotkovými testy jsem přistoupil ke druhé fázi testování, a to k testování integrace komponent.

V této fázi došlo k testování součinnosti jednotlivých procedur a funkcí. Na testovacích datech bylo ověřováno:

- zda po spuštění hlavních procedur dojde k archivaci dat z chybových a importních tabulek podle definovaných kritérií,
- jestli dojde v rámci kontroly konzistence dat ke správnému určení chyb v záznamech,
- zda se tyto chyby správně uloží do chybových tabulek a zda se korektně přeznačí příslušné záznamy v importních tabulkách,
- zda dojde k transformaci dat z importních do finálních tabulek pouze u záznamů se stavem *C* či *WARNING*,
- zda na konci každé hlavní procedury dojde k odeslání e-mailové notifikace,
- jestli přišel správný e-mail správnému uživateli a ve správném formátu,
- zda byly zaznamenány všechny sledované kroky běhu integračního rozhraní v rámci logování.

Všechny tyto testy byly úspěšně splněny.

7.3 Nasazení testovací verze

Po otestování integrace komponent jsem přistoupil ke třetí fázi testování, a to k nasazení testovací verze u zákazníka. Zákazníkem je společnost Futaba Czech, s.r.o., která se zabývá výrobou automobilových dílů. Jedná se o dceřinou firmu japonské společnosti Futaba Industrial Co., Ltd.

Cílem tohoto testu je vyzkoušet funkčnost integračního rozhraní v reálném prostředí, zejména otestovat rychlost běhu a stabilitu při importu velkého objemu dat.

V rámci testovacího provozu je integrační rozhraní spouštěno s reálnými daty zákazníka. Každá tabulka obsahuje v průměru 10 000 záznamů. Z dosavadních výsledků vychází průměrná doba běhu rozhraní přibližně na 2 minuty. Je tedy splněn výkonnostní požadavek na dobu běhu rozhraní (viz kapitola 4.4.1). V rámci testování u zákazníka zatím nebyly zaznamenány chyby a integrační rozhraní funguje bez problémů i při velkém množství dat.

Zákazník je s produktem spokojen, díky rozšiřujícím funkcím může snadno dohledat data s chybami a následně tyto chyby opravit v kmenových datech, o běhu rozhraní je informován prostřednictvím e-mailu, starší data může obnovit z archivních tabulek atd.

Díky rozšiřujícím funkcím integračního rozhraní byly dohledány chybné záznamy a tyto chyby byly opraveny. V průběhu pěti týdnů se podařilo snížit relativní četnost chyb v kmenových datech.

Do budoucna je plánován přechod na reálný provoz rozšířeného integračního rozhraní.

8 Závěr

Cílem této bakalářské práce bylo rozšíření integračního rozhraní vyvíjeného ve společnosti AIMTEC a.s., které zajišťuje přenos dat mezi systémy ERP a APS. V rámci tohoto rozšíření bylo stanoveno několik dílčích cílů.

Prvním dílčím cílem bylo analyzovat problematiku informačních systémů a jejich integrace, dále pak systémy ERP a APS. Díky tomuto cíli jsem získal širší rozhled v oblasti možností integrace informačních systémů, informace o výhodách a nevýhodách jednotlivých metod integrace a obecný pohled na podnikové informační systémy.

Druhým dílčím cílem této práce bylo analyzovat stávající stav integračního rozhraní firmy AIMTEC a.s. mezi ERP a APS systémy a definovat vhodné rozšíření tohoto integračního rozhraní. Na základě analýzy původního rozhraní jsme společně se zadavatelem (firma AIMTEC a.s.) specifikovali požadavky na rozšířené integrační rozhraní. Rozšíření spočívalo v doplnění nových funkcí do stávajícího integračního rozhraní. Na základě specifikace požadavků jsem navrhl strukturu těchto nových funkcí.

Posledním z dílčích cílů byla implementace rozšířeného integračního rozhraní a jeho následné otestování. Na základě návrhu řešení jsem provedl implementaci nových funkcí. V práci charakterizuji technologie a komponenty, které byly při implementaci použity. Následně popisuji průběh testování výsledného řešení, které probíhalo ve třech fázích – nejprve prostřednictvím jednotkových testů, poté byla testována integrace všech komponent a nakonec bylo provedeno nasazení rozšířeného integračního rozhraní do testovacího provozu u zákazníka.

Vzhledem k tomu, že byly splněny všechny dílčí cíle, byl splněn i cíl hlavní. Výstupem je integrační rozhraní, které bylo rozšířeno o kontrolu konzistence dat z pohledu APS systému, archivaci dat, zaslání e-mailových notifikací a logování rozhraní.

V současné době je rozšířené integrační rozhraní nasazené v testovacím provozu ve výrobní společnosti zabývající se výrobou automobilových dílů. Do budoucna je plánován přechod na reálný provoz a nasazení rozšířeného integračního rozhraní u všech zákazníků, kteří mají původní integrační rozhraní od firmy AIMTEC a.s.

Myslím si, že je velký potenciál v dalším rozšíření integračního rozhraní, konkrétně v zavedení kontrol konzistence dat z pohledu ERP systému (při exportu dat z APS do ERP).

Seznam zkratek

API (Application Programming Interface) – neboli aplikační programové rozhraní, je soubor funkcí, procedur, tříd či protokolů aplikace, které může programátor využívat.

APS (Advanced Planning and Scheduling) – neboli pokročilé plánování, je oblast speciálních aplikací v rámci podnikového informačního systému zabývající se výrobním plánováním až po úroveň detailního dílenského rozvrhování.

BI (Business Intelligence) – je manažerský informační systém.

CRM (Customer relationship management) – je řízení vztahu se zákazníkem.

CSS (Cascading Style Sheets) – neboli kaskádové styly, je jazyk popisující vzhled prvků dokumentu, který je psaný ve značkovacím jazyce.

ER (Entity Relationship) – je entitně vztahový model popisující strukturu databáze.

ERP (Enterprise Resource Planning) – neboli plánování podnikových zdrojů, je kategorie podnikových informačních systémů, jimiž podnik za pomoci počítače řídí a integruje všechny nebo většinu oblastí své činnosti (např. plánování, zásoby, nákup, prodej, marketing, finance, personalistika atd.).

ERP II (Enterprise Resource Planning II) – někdy také Extended ERP (rozšířené ERP), jsou komplexní řešení aplikačních software zahrnující a kombinující v sobě funkcionalitu a technologické vlastnosti různých typů aplikací (ERP, CRM, BI, SCM, ...).

ETL (Extract, Transformation, Load) – neboli datové pumpy, jsou nástroje, které se starají o plnění dat do databáze.

FK (Foreign Key) – neboli cizí klíč je v podstatě primární klíč jiné tabulky relační databáze, který slouží k definici vztahů mezi tabulkami.

HTML (Hypertext Markup Language) – je standardní značkovací jazyk používaný zejména pro vytváření webových stránek a webových aplikací.

IBM (International Business Machines Corporation) – je jednou z největších světových společností v oboru informačních technologií.

ICT (Information and Communication Technologies) – neboli informační a komunikační technologie. Jsou do nich zahrnuty veškeré informační technologie používané pro komunikaci a práci s informacemi.

IS (Information System) – neboli informační systém, je soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování, uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení.

PK (Primary Key) – neboli primární klíč, je atribut případně několik atributů tabulky relační databáze, který jednoznačně identifikuje každý záznam.

RAM (Random Access Memory) – je operační paměť počítače, která umožňuje číst nebo zapisovat data do libovolného místa paměti.

SCM (Supply Chain Management) – je řízení dodavatelského řetězce.

SMTP (Simple Mail Transfer Protocol) – je standardní komunikační protokol pro přenos e-mailů na sítích typu TCP/IP.

SQL (Structured Query Language) – je standardizovaný strukturovaný dotazovací jazyk používaný v relačních databázových systémech.

SŘBD (Systém řízení báze dat) – je počítačový systém zabezpečující definování struktury dat, možnost manipulace s daty, ochranu dat a také komunikaci systému s uživatelem.

TCP/IP (Transmission Control Protocol/Internet Protocol) – je sada protokolů, sloužící ke komunikaci v počítačové síti a propojení dvou počítačů.

T-SQL (Transact-SQL) – je proprietární rozšíření jazyka SQL od společností Microsoft a Sybase, které Microsoft používá v produktu Microsoft SQL Server.

Literatura

- [1] BASL, J.; BLAŽÍČEK, R. *Podnikové informační systémy, Podnik v informační společnosti, 3., aktualizované a doplněné vydání*. Grada Publishing, a.s., 2012. ISBN 978-80-247-4307-3.
- [2] BEN-GAN, I.; SARKA, D.; TALMAGE, R. *Querying Microsoft (R) SQL Server (R) 2012, 1. vydání*. Microsoft Press, U.S., 2012. Training Kit (Exam 70-461). ISBN 978-0-7356-6605-4.
- [3] DANEL, R. *Podnikové informační systémy - ERP* [online]. Vysoká škola báňská, 2011. [cit. 2018/12/12]. Informační systémy. Dostupné z: http://home1.vsb.cz/~dan11/is_skripta/IS%202011%20-%20ERP.pdf.
- [4] *Database Mail* [online]. Microsoft Corporation, 2017. [cit. 2019/04/11]. Dostupné z: <https://docs.microsoft.com/en-us/sql/relational-databases/database-mail/database-mail?view=sql-server-2017>.
- [5] GÁLA, L.; POUR, J.; TOMAN, P. *Podniková informatika, 1. vydání*. Grada Publishing, a.s., 2006. ISBN 80-247-1278-4.
- [6] JUŘEK, M. *Moderní integrace aplikací* [online]. Microsoft, s.r.o., 2004. [cit. 2018/12/12]. Dostupné z: http://download.microsoft.com/download/8/6/c/86c09926-affc-4e14-bec0-3c45cd989436/moderni_integrace.pdf.
- [7] MOLNAR, L. *T-SQL Test Tool* [online]. Ladislau Molnar, 2009. [cit. 2019/04/20]. Automating tests for T-SQL code. Dostupné z: <https://archive.codeplex.com/?p=tst>.
- [8] MOLNÁR, Z. *Efektivnost informačních systémů, 2. vydání*. Grada Publishing, a.s., 2001. ISBN 80-247-0087-5.
- [9] ROUSE, M. *Microsoft SQL Server* [online]. TechTarget, Inc., 2017. [cit. 2019/04/09]. Dostupné z: <https://searchsqlserver.techtarget.com/definition/SQL-Server>.
- [10] ROUSE, M. *SQL (Structured Query Language)* [online]. TechTarget, Inc., 2016. [cit. 2019/04/11]. Dostupné z: <https://searchsqlserver.techtarget.com/definition/SQL>.
- [11] ROUSE, M. *T-SQL (Transact-SQL)* [online]. TechTarget, Inc., 2017. [cit. 2019/04/11]. Dostupné z: <https://searchsqlserver.techtarget.com/definition/T-SQL>.

- [12] *Simple Mail Transfer Protocol (SMTP)* [online]. Janalta Interactive Inc., 2016. [cit. 2019/04/12]. Dostupné z: <https://www.techopedia.com/definition/1710/simple-mail-transfer-protocol-smtp>.
- [13] *SQL Server 2017* [online]. Microsoft Corporation, 2017. [cit. 2019/04/09]. Dostupné z: https://www.microsoft.com/en-us/sql-server/sql-server-2017-editions#CP_StickyNav_1.
- [14] *SQL Server Agent* [online]. Microsoft Corporation, 2017. [cit. 2019/04/11]. Dostupné z: <https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent?view=sql-server-2017>.
- [15] *T-SQL vs SQL* [online]. Database.Guide, 2018. [cit. 2019/04/11]. Dostupné z: <https://database.guide/t-sql-vs-sql/>.
- [16] *What is CSS?* [online]. W3Schools, 2016. [cit. 2019/04/12]. CSS Introduction. Dostupné z: https://www.w3schools.com/css/css_intro.asp.
- [17] *What is HTML?* [online]. W3Schools, 2016. [cit. 2019/04/12]. HTML Introduction. Dostupné z: https://www.w3schools.com/html/html_intro.asp.
- [18] *What is Unit Testing?* [online]. Guru99, 2011. [cit. 2019/04/13]. Unit Testing Tutorial: What is, Types, Tools, EXAMPLE. Dostupné z: <https://www.guru99.com/unit-testing-guide.html>.

Seznam příloh

A	Tabulky původního rozhraní	59
A.1	Názvy tabulek – transformace z ERP do APS	59
A.2	Struktura tabulky Item	60
B	Procedury původního rozhraní	61
C	Běh původního rozhraní	62
D	Tabulky rozšířeného rozhraní	63
D.1	Názvy tabulek – transformace z ERP do APS	63
D.2	Společné atributy importních tabulek	64
D.3	Atributy chybové tabulky pro položky	64
D.4	Atributy logovací tabulky	65
D.5	Atributy tabulky e-mailových notifikací	65
E	Procedury rozšířeného rozhraní	66
E.1	Názvy procedur – transformační a kontrolní procedury	66
E.2	Hierarchie notifikačních procedur	67
F	Běh rozšířeného rozhraní	68
G	Kontroly pro položky	70
H	Záznamy logu pro položky	71
I	Uživatelská dokumentace	73
I.1	Vytvoření instalátoru	73
I.2	Instalace databázové aplikace	74
I.2.1	Vytvoření základní databáze	75
I.2.2	Instalace databázových skriptů	77
I.3	Konfigurace databázové aplikace	78
I.3.1	Nastavení globálních proměnných	78
I.3.2	Nastavení SQL jobů	78
I.3.3	Nastavení e-mailových notifikací	79
I.4	Spuštění databázové aplikace	81
I.5	Vstupy a výstupy databázové aplikace	82

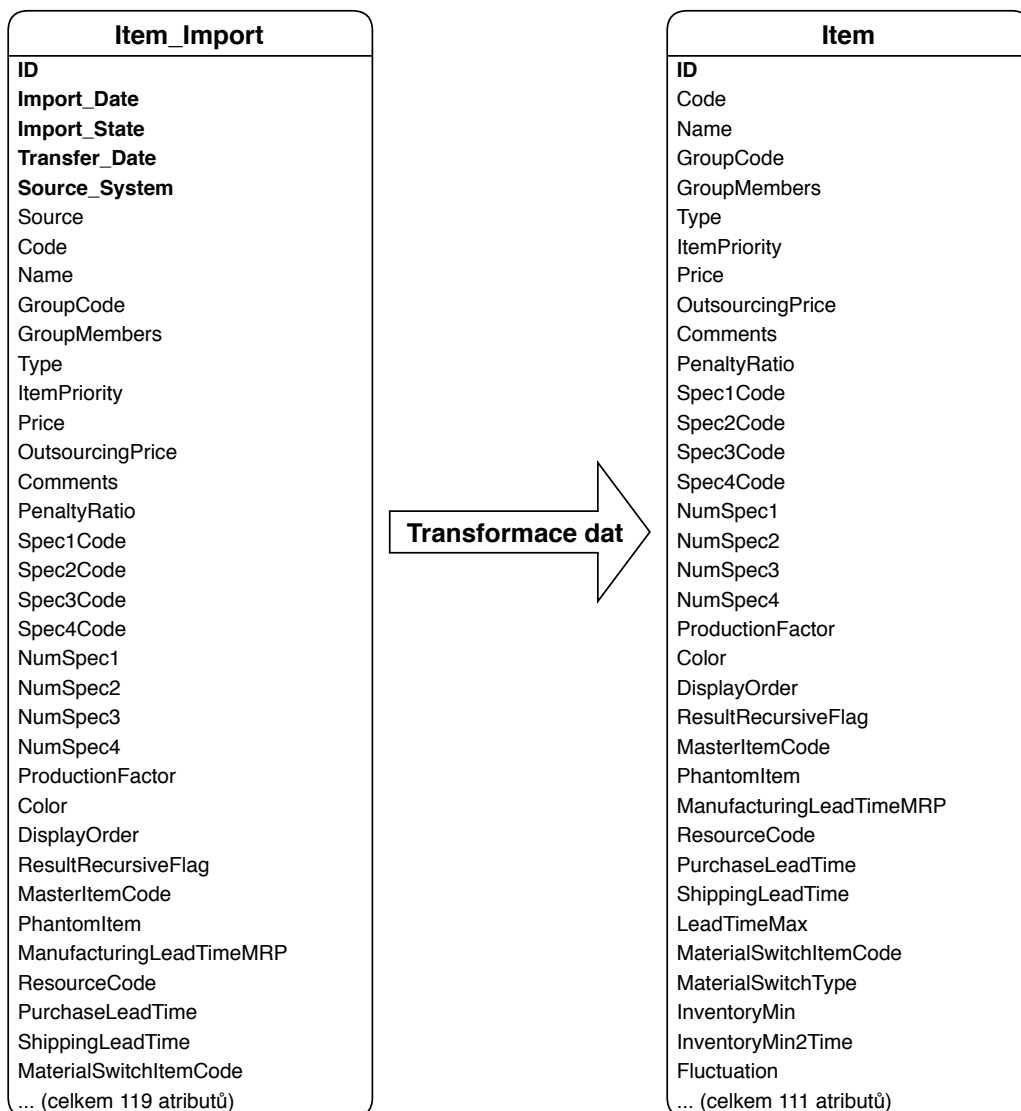
J	Obsah CD	84
J.1	Struktura obsahu CD	84
J.2	Popis obsahu CD	84

A Tabulky původního rozhraní

A.1 Názvy tabulek – transformace z ERP do APS

Tabulka z kapitoly 3.3.1	Tabulka v rozhraní		Popis
	Importní	Finální	
Položky	Item_Import	Item	Tabulka položek
Zdroje	Resource_Import	Resource	Tabulka zdrojů
Operace	Operation_Import	Operation	Tabulka operací
Zakázky	Order_Import	Order	Tabulka zakázek
Zákazníci	Customer_Import	Customer	Tabulka zákazníků
Směny	Shift_Import	Shift	Tabulka směn
Kalendáře	Calendar_Import	Calendar	Tabulka kalendářů
Kusovník s postupy	IM_In_Import	IM_In	Tabulka vstupů
Kusovník s postupy	IM_OUT_Import	IM_OUT	Tabulka výstupů
Kusovník s postupy	IM_Use_M_Import	IM_Use_M	Tabulka výrobních postupů
Kusovník s postupy	IM_Use_Sx_Import	IM_Use_Sx	Tabulka pomocných výrobních prostředků

A.2 Struktura tabulky Item



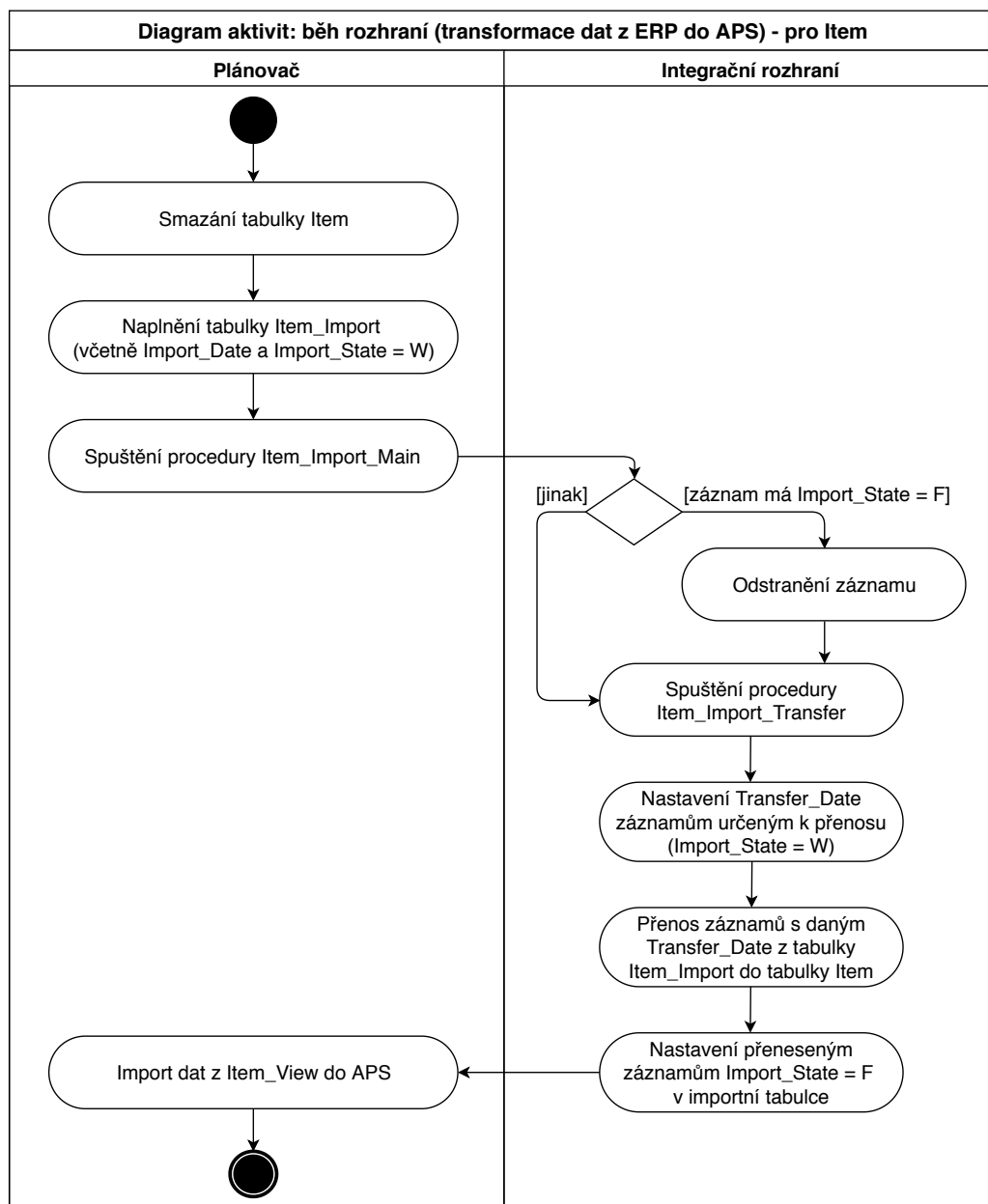
B Procedury původního rozhraní

Názvy procedur – transformace z ERP do APS

Importní tabulka	Hlavní procedura	Procedura pro přenos	Vytvoření pohledu
Item_Import	Item_Import_Main	Item_Import_Transfer	Create_Item_View
Resource_Import	Resource_Import_Main	Resource_Import_Transfer	Create_Resource_View
Operation_Import	Operation_Import_Main	Operation_Import_Transfer	Create_Operation_View
Order_Import	Order_Import_Main	Order_Import_Transfer	Create_Order_View
Customer_Import	Customer_Import_Main	Customer_Import_Transfer	Create_Customer_View
Shift_Import	Shift_Import_Main	Shift_Import_Transfer	Create_Shift_View
Calendar_Import	Calendar_Import_Main	Calendar_Import_Transfer	Create_Calendar_View
IM_In_Import	IM_In_Import_Main	IM_In_Import_Transfer	Create_IM_In_View
IM_OUT_Import	IM_OUT_Import_Main	IM_OUT_Import_Transfer	Create_IM_OUT_View
IM_Use_M_Import	IM_Use_M_Import_Main	IM_Use_M_Import_Transfer	Create_IM_Use_M_View
IM_Use_Sx_Import	IM_Use_Sx_Import_Main	IM_Use_Sx_Import_Transfer	Create_IM_Use_Sx_View

C Běh původního rozhraní

Transformace dat z ERP do APS



D Tabulky rozšířeného rozhraní

D.1 Názvy tabulek – transformace z ERP do APS

Tabulka z kap. 3.3.1	Tabulka v rozhraní			Popis
	Importní	Chybová	Finální	
Položky	Item_Import	Item_Import_Error	Item	Tabulka položek
Zdroje	Resource_Import	Resource_Import_Error	Resource	Tabulka zdrojů
Operace	Operation_Import	Operation_Import_Error	Operation	Tabulka operací
Zakázky	Order_Import	Order_Import_Error	Order	Tabulka zakázek
Zákazníci	Customer_Import	Customer_Import_Error	Customer	Tabulka zákazníků
Směny	Shift_Import	Shift_Import_Error	Shift	Tabulka směn
Kalendáře	Calendar_Import	Calendar_Import_Error	Calendar	Tabulka kalendářů
Kusovník s postupy	IM_In_Import	IM_In_Import_Error	IM_In	Tabulka vstupů
Kusovník s postupy	IM_OUT_Import	IM_OUT_Import_Error	IM_OUT	Tabulka výstupů
Kusovník s postupy	IM_Use_M_Import	IM_Use_M_Import_Error	IM_Use_M	Tabulka výrobních postupů
Kusovník s postupy	IM_Use_Sx_Import	IM_Use_Sx_Import_Error	IM_Use_Sx	Tabulka pomocných výrobních prostředků

D.2 Společné atributy importních tabulek

Název atributu	Datový typ	Popis	Klíč	Null
ID	integer	identifikátor záznamu	PK	Ne
Import_Date	datetime	datum importu záznamu	–	Ne
Import_State	nvarchar(10)	aktuální importní stav záznamu	–	Ne
Import_Error_Code	nvarchar(max)	kódové označení chyby nebo chyb	–	Ano
Import_Error_Reason	nvarchar(max)	popis příčiny chyby nebo chyb	–	Ano
Transfer_Date	datetime	datum přenosu záznamu z importní tabulky do finální tabulky	–	Ano
Check_Date	datetime	datum kontroly záznamu	–	Ano
Source_System	integer	systém, ze kterého jsou importovány záznamy	–	Ne
Is_Notified	bit	příznak, zda byl chybový záznam v importní tabulce notifikován	–	Ano

D.3 Atributy chybové tabulky pro položky

Název atributu	Datový typ	Popis	Klíč	Null
ID	integer	identifikátor záznamu	PK	Ne
Item_Import_ID	integer	identifikátor záznamu v importní tabulce	FK	Ne
Item_Import_Import_Date	datetime	datum importu odpovídajícího záznamu v importní tabulce	–	Ne
Import_State	nvarchar(10)	aktuální stav chybového záznamu	–	Ne
Import_Error_Code	nvarchar(6)	kódové označení dané chyby	–	Ne
Import_Error_Reason	nvarchar(max)	popis příčiny dané chyby	–	Ne

D.4 Atributy logovací tabulky

Název atributu	Datový typ	Popis	Klíč	Null
ID	integer	identifikátor záznamu	PK	Ne
Log_Number	integer	číslo kategorie záznamu v logu	–	Ano
Table	nvarchar(255)	název tabulky, ke které daný záznam v logu náleží	–	Ano
Time	datetime	datum vytvoření záznamu v logu	–	Ano
Message	nvarchar(4000)	zpráva k danému záznamu v logu	–	Ano
Comment	nvarchar(4000)	komentář k danému záznamu v logu	–	Ano

D.5 Atributy tabulky e-mailových notifikací

Název atributu	Datový typ	Popis	Klíč	Null
ID	integer	identifikátor záznamu	PK	Ne
Table_Name	nvarchar(100)	název <i>importní</i> nebo <i>logovací</i> tabulky	–	Ano
Notified_Column_Names	nvarchar(max)	názvy sloupců, které mají být notifikovány	–	Ano
To_Address	nvarchar(max)	adresa příjemce	–	Ano
From_Address	nvarchar(max)	adresa odesílatele	–	Ano
Subject	nvarchar(255)	předmět e-mailu	–	Ano
Comment	nvarchar(max)	komentář k e-mailu	–	Ano
Role	nvarchar(255)	role pro daný záznam	–	Ano
Log_Time_Offset	integer	časový úsek ve dnech, pro který má být odeslán log	–	Ano

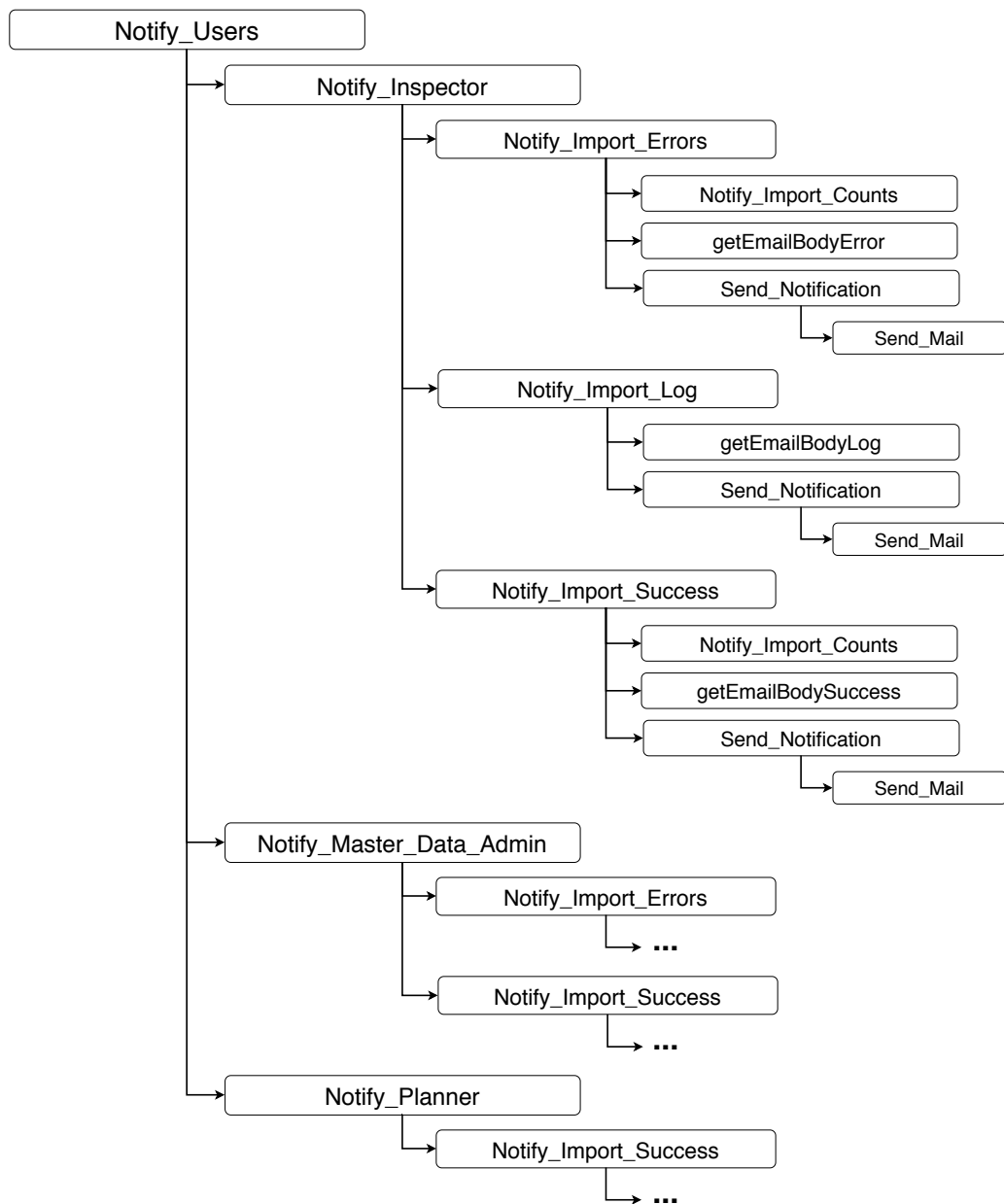
E Procedury rozšířeného rozhraní

E.1 Názvy procedur – transformační a kontrolní procedury

Importní tabulka	Hlavní procedura	Kontrolní procedura	
Item_Import	Item_Import_Main	Item_Import_Check	
Resource_Import	Resource_Import_Main	Resource_Import_Check	
Operation_Import	Operation_Import_Main	Operation_Import_Check	
Order_Import	Order_Import_Main	Order_Import_Check	
Customer_Import	Customer_Import_Main	Customer_Import_Check	
Shift_Import	Shift_Import_Main	Shift_Import_Check	
Calendar_Import	Calendar_Import_Main	Calendar_Import_Check	
IM_In_Import	IM_In_Import_Main	IM_In_Import_Check	
IM_OUT_Import	IM_OUT_Import_Main	IM_OUT_Import_Check	
IM_Use_M_Import	IM_Use_M_Import_Main	IM_Use_M_Import_Check	
IM_Use_Sx_Import	IM_Use_Sx_Import_Main	IM_Use_Sx_Import_Check	

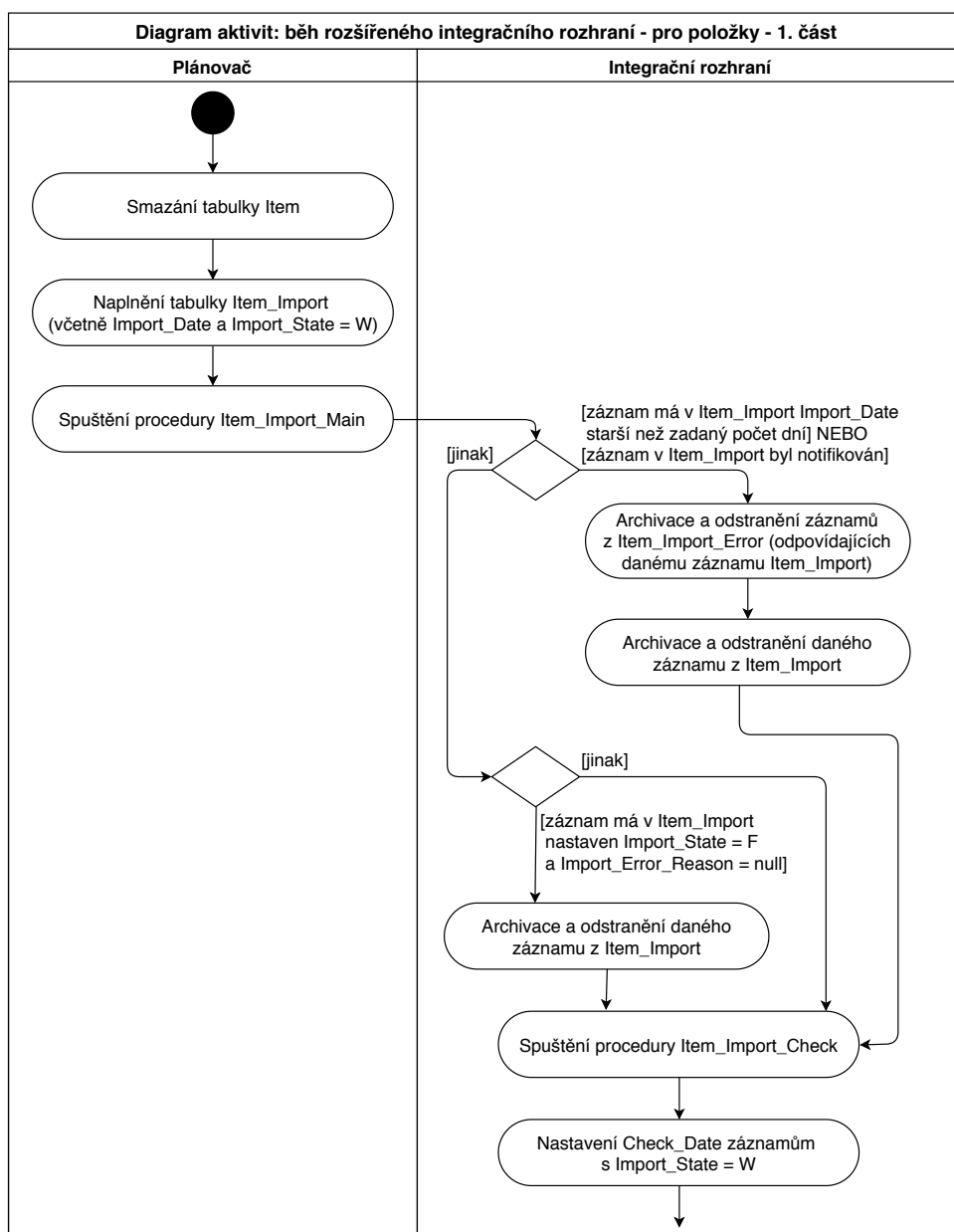
	Vyhodnocovací procedura	Procedura pro přenos
	Item_Import_Check_Evaluation_State	Item_Import_Transfer
	Resource_Import_Check_Evaluation_State	Resource_Import_Transfer
	Operation_Import_Check_Evaluation_State	Operation_Import_Transfer
	Order_Import_Check_Evaluation_State	Order_Import_Transfer
	Customer_Import_Check_Evaluation_State	Customer_Import_Transfer
	Shift_Import_Check_Evaluation_State	Shift_Import_Transfer
	Calendar_Import_Check_Evaluation_State	Calendar_Import_Transfer
	IM_In_Import_Check_Evaluation_State	IM_In_Import_Transfer
	IM_OUT_Import_Check_Evaluation_State	IM_OUT_Import_Transfer
	IM_Use_M_Import_Check_Evaluation_State	IM_Use_M_Import_Transfer
	IM_Use_Sx_Import_Check_Evaluation_State	IM_Use_Sx_Import_Transfer

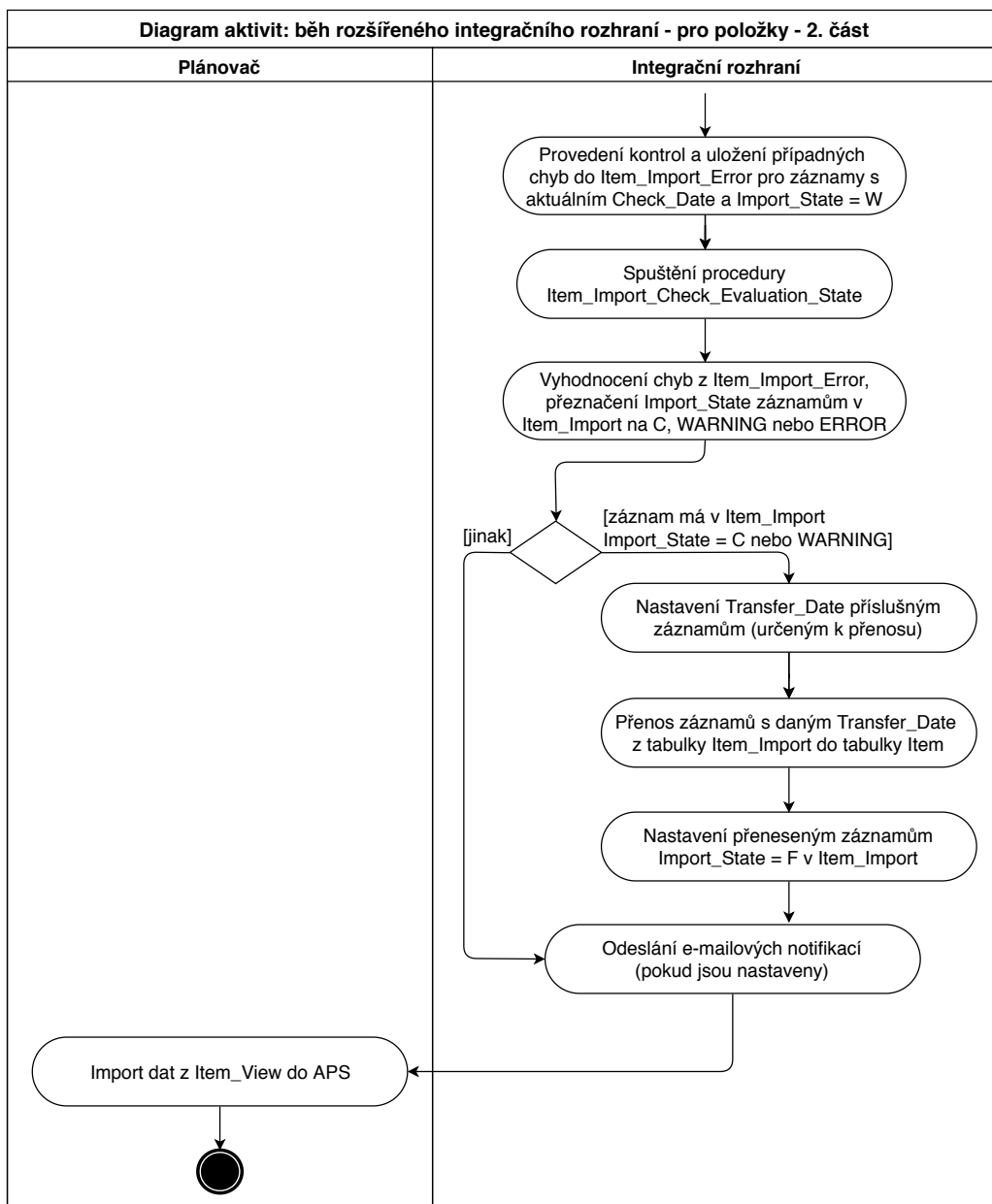
E.2 Hierarchie notifikačních procedur



F Běh rozšířeného rozhraní

Transformace dat z ERP do APS včetně kontroly konzistence dat





G Kontroly pro položky

Typ chyby	Kód chyby	Příčina chyby
ERROR	EIT001	Item code is missing. Fill in the field "Code".;
WARNING	WIT001	Value in the field "ManufacturingOrderQtyMin" is bigger than the value in the field "ManufacturingOrderQtyMax". Redefine value in ManufacturingOrderQtyMin and/or ManufacturingOrderQtyMax.;
WARNING	WIT002	Value in the field "ManufacturingOrderQtyUnit" is bigger than value in the field "ManufacturingOrderQtyMax". Redefine value in ManufacturingOrderQtyUnit and/or ManufacturingOrderQtyMax.;
WARNING	WIT003	Value in the field "PurchaseOrderQtyMin" is bigger than the value in the field "PurchaseOrderQtyMax". Redefine value in PurchaseOrderQtyMin and/or PurchaseOrderQtyMax.;
WARNING	WIT004	Value in the field "PurchaseOrderQtyUnit" is bigger than value in the field "PurchaseOrderQtyMax". Redefine value in ManufacturingOrderQtyUnit and/or ManufacturingOrderQtyMax.;
WARNING	WIT005	Item type is missing or is wrongly defined. Fill in the field "Type" with one of the standard Item type: M (Raw material), U (Purchased part), H (Manufactured part), I (Intermediate item), P (Finished item).;
WARNING	WIT006	Item name is missing. Fill in field "Name".;
WARNING	WIT007	Obtain method is missing. Choose one of the standard Obtain method: N (Not allowed), M, (Produce), P (Purchase), IM (Prefer to produce), IP (Prefer to purchase).;

H Záznamy logu pro položky

Kat.	Zpráva (message)	Komentář (comment)	Popis
0	Deleted - table Item.	Records from the table Item have been deleted.	Odstraněny záznamy z tabulky Item.
1	Inserted - table Item_Import.	Records to the table Item_Import have been inserted.	Vloženy záznamy do tabulky Item_Import.
2	Started - Item_Import_Archive and Item_Import_Error_Archive.	Archiving of Item_Import and Item_Import_Error has started.	Začátek archivace tabulek Item_Import a Item_Import_Error.
	Finished - Item_Import_Archive and Item_Import_Error_Archive.	No items for archiving.	Konec archivace tabulek Item_Import a Item_Import_Error, není co k archivaci.
	Finished - Item_Import_Archive and Item_Import_Error_Archive.	Archiving of Item_Import and Item_Import_Error has successfully finished.	Konec archivace tabulek Item_Import a Item_Import_Error, úspěšně dokončeno.
3	Started - Item_Import_Check.	Checking of Item_Import has started.	Začátek kontrolní procedury pro Item_Import.
	Finished - Item_Import_Check.	No items for checking.	Konec kontrolní procedury pro Item_Import, není co kontrolovat.
	Started - Item_Import_Check_Evaluation_State.	Evaluating Import_State of Item_Import has started.	Začátek vyhodnocovací procedury pro Item_Import.
	Finished - Item_Import_Check_Evaluation_State.	Evaluating Import_State of Item_Import successfully finished.	Konec vyhodnocovací procedury pro Item_Import, úspěšně dokončeno.
	Finished - Item_Import_Check.	Checking of Item_Import has successfully finished.	Konec kontrolní procedury pro Item_Import, úspěšně dokončeno.
4	Started - Item_Import_Transfer.	Transferring from Item_Import to Item has started.	Začátek procedury pro přenos z Item_Import do Item.
	Finished - Item_Import_Transfer.	No items for transferring.	Konec procedury pro přenos z Item_Import do Item, není co přenášet.
	Finished - Item_Import_Transfer.	Transferring from Item_Import to Item has successfully finished.	Konec procedury pro přenos z Item_Import do Item, úspěšně dokončeno.
	Failed - Item_Import_Transfer.	Transferring from Item_Import to Item has failed.	Přenos z tabulky Item_Import do tabulky Item selhal.

pokračování na další stránce

pokračování tabulky přílohy H

Kat.	Zpráva (message)	Komentář (comment)	Popis
5	Started - Notify_Users.	Sending notifications has started.	Začátek hlavní notifikační procedury Notify_Users.
	Finished - Notify_Users.	Sending notifications has successfully finished.	Konec hlavní notifikační procedury Notify_Users, úspěšně dokončeno.
	Failed - Notify_Users.	Sending notifications has failed.	Hlavní notifikační procedura Notify_Users selhala.
	Started - Notify_Inspector.	Sending notifications has started.	Začátek notifikační procedury Notify_Inspector.
	Finished - Notify_Inspector.	Sending notifications has finished.	Konec notifikační procedury Notify_Inspector.
	Started - Notify_Master_Data_Admin.	Sending notifications has started.	Začátek notifikační procedury Notify_Master_Data_Admin.
	Finished - Notify_Master_Data_Admin.	Sending notifications has finished.	Konec notifikační procedury Notify_Master_Data_Admin.
	Started - Notify_Planner.	Sending notifications has started.	Začátek notifikační procedury Notify_Planner.
	Finished - Notify_Planner.	Sending notifications has finished.	Konec notifikační procedury Notify_Planner.
	Started - Notify_Import_Errors.	Sending import error notifications has started.	Začátek notifikační procedury Notify_Import_Errors.
	Finished - Notify_Import_Errors.	Sending import error notifications has finished.	Konec notifikační procedury Notify_Import_Errors.
	Started - Notify_Import_Log.	Sending log has started.	Začátek notifikační procedury Notify_Import_Log.
	Finished - Notify_Import_Log.	Sending log has finished.	Konec notifikační procedury Notify_Import_Log.
	Started - Notify_Import_Success.	Sending success has started.	Začátek notifikační procedury Notify_Import_Success.
	Finished - Notify_Import_Success.	Sending success has finished.	Konec notifikační procedury Notify_Import_Success.
	99	Started - Archive_Item.	Archiving of Item has started.
Finished - Archive_Item.		No items for archiving.	Konec archivace tabulky Item, není co archivovat.
Finished - Archive_Item.		Archiving of Item has successfully finished.	Konec archivace tabulky Item, úspěšně dokončeno.
Failed - Archive_Item.		Archiving of Item has failed.	Archivace tabulky Item selhala.

I Uživatelská dokumentace

V následující kapitole je popsáno vytvoření instalátoru, instalace databázové aplikace, její konfigurace, spuštění, vstupy a výstupy. Aplikace je určena pro prostředí operačních systémů Microsoft Windows, proto je všechen postup uváděn pro tyto operační systémy.

I.1 Vytvoření instalátoru

Aby bylo možné vytvořit instalátor integračního rozhraní se všemi databázovými skripty, je nutné mít *nainstalované* a *správně nastavené* následující softwarové vybavení:

- Java Development Kit (JDK) – verze 8 a vyšší,
- Apache Maven – verze 3.3 a vyšší,
- Apache Ant – verze 1.9.14 a vyšší,
- Nullsoft Scriptable Install System (NSIS) – verze 3.0 a vyšší.

Pokud je softwarové vybavení nainstalované a správně nastavené, je možné vytvořit instalátor. Vytvoření instalátoru je automatizováno nástrojem **Maven**, který využívá konfigurační soubor `pom.xml` v adresáři **Aplikace**. Překlad se spouští přes Příkazový řádek.

Postup vytvoření instalátoru:

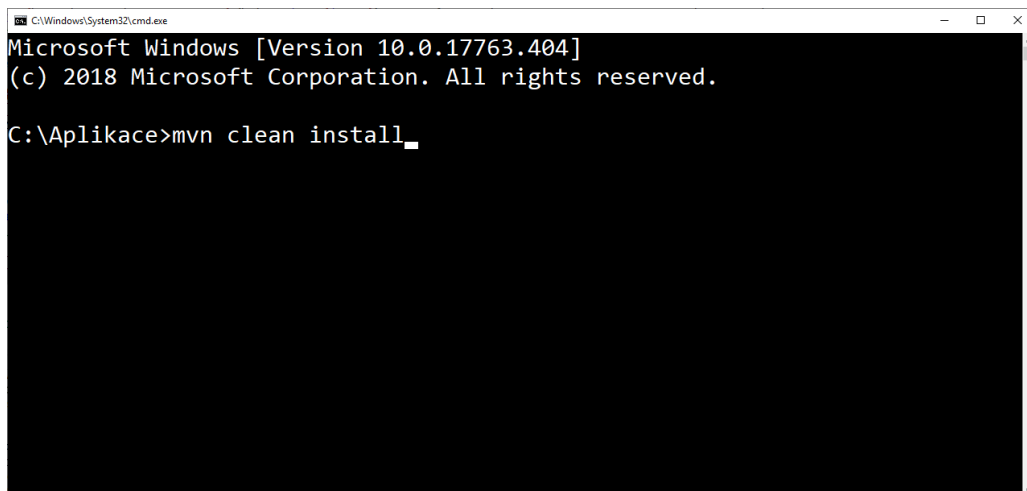
1. otevřít Příkazový řádek,
2. změnit aktuální cestu do adresáře **Aplikace**,
3. zadat příkaz: `mvn clean install` (viz obrázek I.1).

Pokud je vytvoření instalátoru úspěšné, je v Příkazovém řádku zobrazena informace `BUILD SUCCESS` (viz obrázek I.2). Instalátor je spustitelný soubor s názvem:

```
ASP_Integration_2.0.0.exe,
```

a nachází se v umístění:

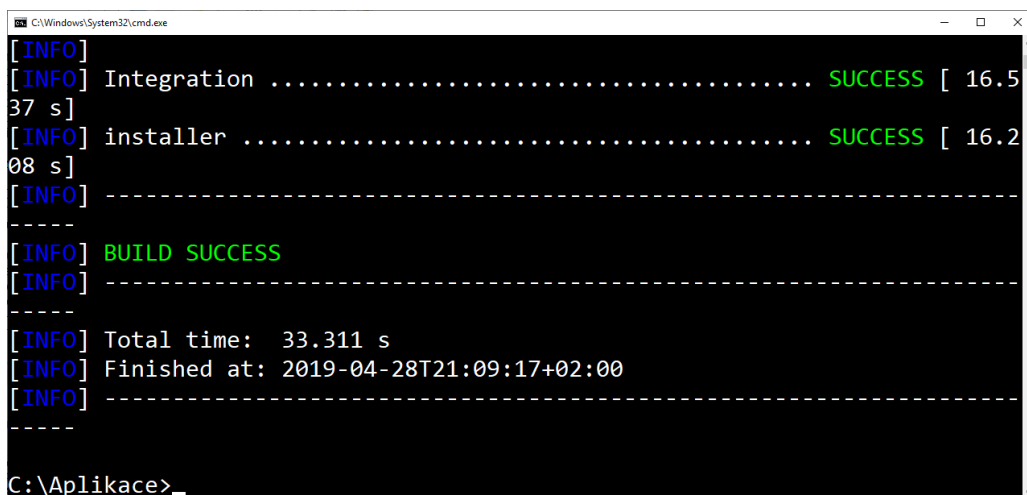
```
Aplikace\installer\target\scripts\nsis.
```



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17763.404]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Aplikace>mvn clean install_
```

Obrázek I.1: Příkaz pro vytvoření instalátoru databázových skriptů



```
C:\Windows\System32\cmd.exe
[INFO]
[INFO] Integration ..... SUCCESS [ 16.5
37 s]
[INFO] installer ..... SUCCESS [ 16.2
08 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 33.311 s
[INFO] Finished at: 2019-04-28T21:09:17+02:00
[INFO] -----
C:\Aplikace>
```

Obrázek I.2: Úspěšné vytvoření instalátoru databázových skriptů

I.2 Instalace databázové aplikace

Před instalací databázové aplikace je nutné zabezpečit, aby v provozním prostředí této aplikace byl nainstalován a správně nastaven Microsoft SQL Server 2016 Standard Edition SP1 a vyšší. Pro přístup k SQL Serveru je doporučeno použít grafické uživatelské rozhraní Microsoft SQL Server Management Studio 2016 a vyšší. Následující postup bude uveden právě s použitím tohoto grafického rozhraní.

I.2.1 Vytvoření základní databáze

Aby bylo možné nainstalovat databázovou aplikaci, je nutné nejprve vytvořit *základní databázi* v prostředí Microsoft SQL Serveru. Vytvoření probíhá tak, že se obnoví ze zálohy již předem vytvořená základní databáze. Tato základní databáze je zabalena v archivu `.zip` s názvem:

`asp_core.zip`,

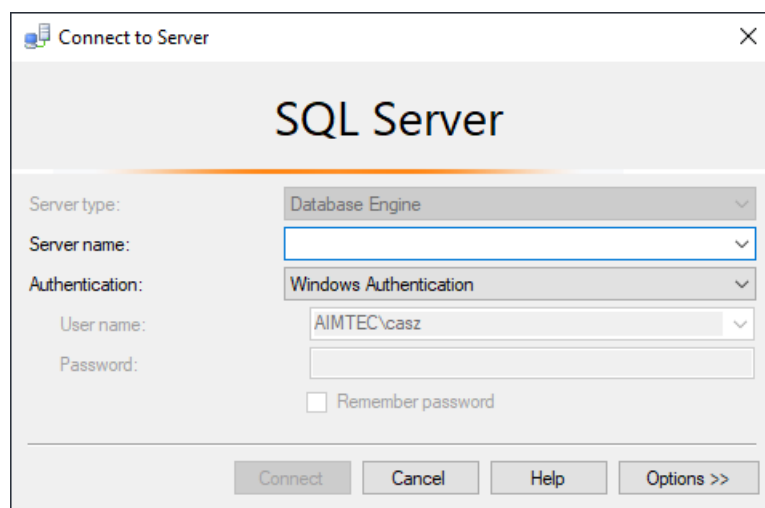
a je uložena v umístění:

`Aplikace\src\main\resources\DB.`

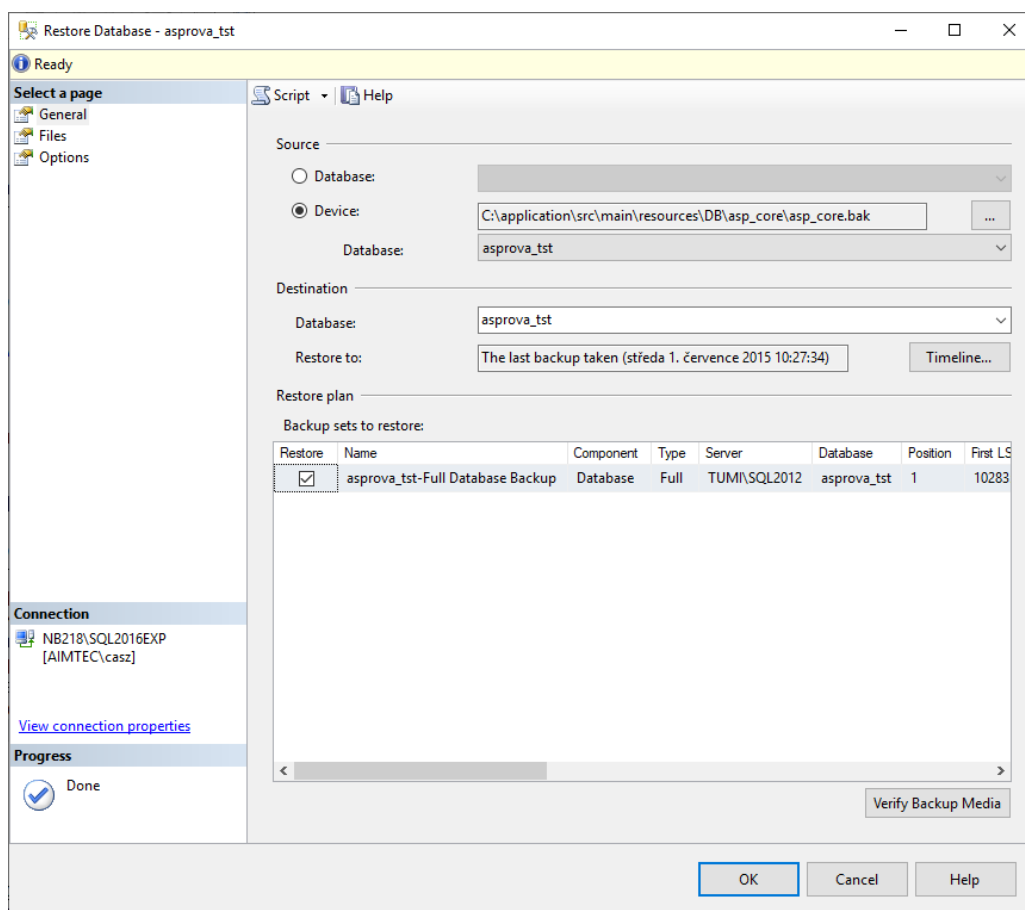
Možný postup obnovy základní databáze ze zálohy:

1. extrahovat základní databázi `asp_core.zip` (soubor se zálohou je `asp_core.bak`),
2. otevřít Microsoft SQL Server Management Studio,
3. připojit se k databázovému serveru vyplněním údajů v dialogovém okně `Connect to Server` (viz obrázek I.3),
4. kliknout pravým tlačítkem myši na složku `Databases` v sekci `Object Explorer`, poté kliknout na `Restore Database...`,
5. v dialogovém okně `Restore Database` zvolit v záložce `General` v sekci `Source` možnost `Device` a vpravo kliknout na tlačítko `...`,
6. v dialogovém okně `Select backup devices` nastavit `Backup media type` na hodnotu `File` a kliknout na tlačítko `Add`,
7. v okně `Locate Backup File` vyhledat soubor se *základní databází* `asp_core.bak` a kliknout na tlačítko `OK`,
8. v okně `Select backup devices` potvrdit kliknutím na tlačítko `OK`,
9. zvolit název databáze v okně `Restore Database`, např. `asprova_tst` (dialogové okno by mělo vypadat podobně jako na obrázku I.4),
10. potvrdit tlačítkem `OK`.

Výše uvedený postup popisuje pouze základní nastavení při obnově databáze, které je ale v tomto případě dostačující. Po obnově je základní databáze umístěna ve složce `Databases` a obsahuje pouze jednu tabulku `dbo.Admin_Scripts`.



Obrázek I.3: Připojení k databázovému serveru



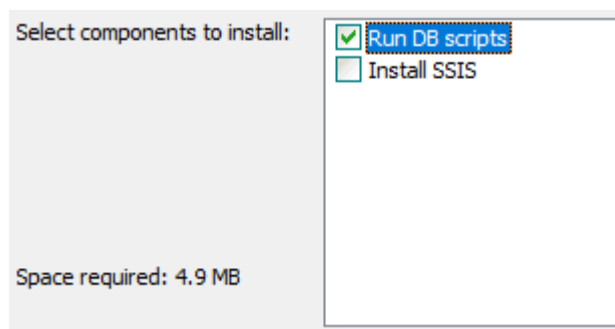
Obrázek I.4: Vytvoření základní databáze

I.2.2 Instalace databázových skriptů

Nyní je možné nainstalovat databázové skripty do základní databáze. Instalace je realizována prostřednictvím instalátoru databázových skriptů, jehož vytvoření je popsáno v kapitole I.1.

Možný postup instalace databázových skriptů:

1. spustit instalátor `ASP_Integration_2.0.0.exe`,
2. v sekci **Select components to install** vybrat pouze **Run DB scripts** (viz obrázek I.5) a poté kliknout na tlačítko **Next >**,
3. vyplnit **Database server name** (název databázového serveru), **Database name** (název základní databáze, např. `asprova_tst`), **User** (uživatelské jméno pro přihlášení k databázovému serveru), **Password** (heslo),
4. kliknout na tlačítko **Next >**,
5. potvrdit dialogové okno **Connection OK** tlačítkem **OK** (v případě, že byly správně vyplněny všechny údaje a že je spojení úspěšné),
6. instalace je spuštěna (pokud je instalace úspěšná, v instalačním okně se zobrazí informace **Completed**),
7. po dokončení instalace kliknout na **Close**.



Obrázek I.5: Instalace databázových skriptů

I.3 Konfigurace databázové aplikace

Po instalaci databázové aplikace je nutné ji správně nakonfigurovat. Je třeba nastavit *hodnoty globálních proměnných*, vytvořit a nastavit *SQL joby* a nakonfigurovat *odesílání e-mailových notifikací*.

I.3.1 Nastavení globálních proměnných

Pro nastavení hodnot globálních proměnných je ve zdrojových souborech vytvořen skript s názvem:

```
Add.Integration_Properties.sql,
```

a nachází se v umístění:

```
Aplikace\src\main\scripts\DB\usefulScripts.
```

V kódu tohoto skriptu jsou nastaveny výchozí hodnoty globálních proměnných, v případě potřeby je možné je změnit. Za parametr `@dbName` je nutné zadat název databáze integračního rozhraní, pro kterou chceme nastavit hodnoty proměnných (např. `asprova_tst`). Skript je nutné spouštět pod danou databází integračního rozhraní (např. `asprova_tst`). Po spuštění tohoto skriptu pod správnou databází dojde k nastavení hodnot globálních proměnných.

I.3.2 Nastavení SQL jobů

Nastavení SQL jobů se skládá ze dvou částí: *založení SQL jobů* a jejich *aktivace*.

Založení SQL jobů

Pro založení SQL jobů jsou ve zdrojových souborech vytvořeny tři skripty s názvy:

```
Create.job.Archive_Clean_Tables.sql,
```

```
Create.job.CUST_Archive_Tables_Maintenance.sql,
```

```
Create.job.Table_Maintenance.sql,
```

a nachází se v umístění:

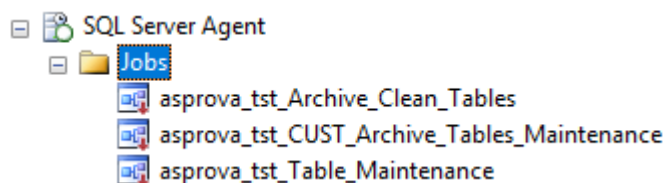
```
Aplikace\src\main\scripts\DB\usefulScripts.
```

Skript `Create.job.Archive_Clean_Tables.sql` založí job ke spuštění archivace finálních tabulek, skript `Create.job.CUST_Archive_Tables_Maintenance.sql` založí job k zajištění periodického odstraňování starých záznamů z archivních importních a archivních chybových tabulek a skript `Create.job.Table_Maintenance.sql` založí job k periodickému odstraňování starých záznamů z archivních finálních tabulek.

V těchto skriptech je doporučeno upravit proměnnou `@tst` na název databáze, pro kterou chceme SQL joby založit (např. `asprova_tst`). Dále je možné nastavit název SQL jobu a periodu opakování jobu přímo ve skriptu. Skripty je nutné spouštět pod databází integračního rozhraní (např. `asprova_tst`). Po spuštění těchto skriptů pod správnou databází se založí příslušné joby, ovšem se stavem `Disabled` (deaktivovaný). V dalším kroku je tedy nutné je aktivovat.

Aktivace SQL jobů

Založené SQL joby jsou zobrazeny v sekci `Object Explorer` v komponentě `SQL Server Agent` ve složce `Jobs` (viz obrázek I.6).



Obrázek I.6: Založené SQL joby

Pro aktivaci daného SQL jobu stačí kliknout pravým tlačítkem myši na daný job a zvolit možnost `Enable` (aktivovat). Také je možné nastavit více parametrů daného jobu, např. změnit periodu opakování. K tomu slouží volba `Properties`, která se zobrazí po kliknutí pravým tlačítkem myši na příslušný job.

I.3.3 Nastavení e-mailových notifikací

V rámci nastavení e-mailových notifikací je nutné nejprve *vytvořit e-mailový profil a účet* a následně *vyplnit tabulku `Email_Notifications`*.

Vytvoření e-mailového profilu a účtu

Pro vytvoření e-mailového profilu a účtu je ve zdrojových souborech vytvořen skript s názvem:

```
Create.ASP.Sysmail.Account.sql,
```

a nachází se v umístění:

```
Aplikace\src\main\scripts\DB\usefulScripts.
```

V tomto skriptu je nutné vyplnit proměnné `@email_address` (výchozí e-mailová adresa odesílatele) a `@mailserver_name` (adresa SMTP serveru odesílatele). V případě, že SMTP server vyžaduje autentifikaci, je nutné vyplnit proměnné `@username` (uživatelské jméno) a `@password` (heslo). Dále je možné nastavit port SMTP serveru (`@port`), název účtu (`@account_name`) atd.

Po spuštění tohoto skriptu dojde k vytvoření definovaného e-mailového profilu a e-mailového účtu.

Vyplnění tabulky `Email_Notifications`

K tomu, aby bylo možné odesílat e-mailové notifikace, musí být vyplněna tabulka `Email_Notifications` požadovanými informacemi. Atributy této tabulky jsou popsány v kapitole 5.6.2.

Pro předvyplnění tabulky `Email_Notifications` je ve zdrojových souborech vytvořen skript s názvem:

```
Fill.Email.Notifications.Table.sql,
```

a nachází se v umístění:

```
Aplikace\src\main\scripts\DB\usefulScripts.
```

Tento skript vyžaduje vyplnění proměnné `@toAddress` (e-mailová adresa příjemce). Je možné vyplnit proměnnou `fromAddress` (e-mailová adresa odesílatele), pokud nevyhovuje výchozí e-mailová adresa odesílatele.

Po spuštění skriptu dojde k předvyplnění tabulky `Email_Notifications`. Adresa příjemce `@toAddress` je pro všechny role uživatelů stejná. Je tedy vhodné ji po spuštění skriptu upravit v tabulce `Email_Notifications` v závislosti na příslušné roli uživatele.

I.4 Spuštění databázové aplikace

Jelikož se rozšíření integračního rozhraní v rámci této bakalářské práce týká pouze *transformace datových struktur z ERP do APS*, je v této kapitole uvedeno spuštění databázové aplikace pouze pro tuto část.

Před spuštěním je nutné *odstranit záznamy z finálních tabulek* a *vložit nové záznamy do importních tabulek* s importním stavem (`Import_State`) *W* a aktuálním datem importu (`Import_Date`). Na pořadí těchto dvou operací nezáleží.

Poté je možné spustit databázovou aplikaci. Spuštění aplikace se realizuje sekvenčním vykonáním *hlavních procedur* integračního rozhraní v definovaném pořadí.

Volání hlavních procedur ve správném pořadí:

1. `exec Item_Import_Main`
2. `exec Resource_Import_Main`
3. `exec Customer_Import_Main`
4. `exec IM_Use_M_Import_Main`
5. `exec IM_OUT_Import_Main`
6. `exec IM_In_Import_Main`
7. `exec IM_Use_Sx_Import_Main`
8. `exec Order_Import_Main`
9. `exec Operation_Import_Main`
10. `exec Shift_Import_Main`
11. `exec Calendar_Import_Main`

Pro spuštění databázové aplikace je také možné využít skript s názvem:

```
Import.Script.sql,
```

který se nachází v umístění:

```
Aplikace\src\main\scripts\DB\usefulScripts.
```

Před použitím toho skriptu je nutné vložit nové záznamy do importních tabulek se správným importním stavem a datem importu. Skript zajistí odstranění záznamů z finálních tabulek a sekvenční spuštění hlavních procedur.

I.5 Vstupy a výstupy databázové aplikace

V rámci transformace struktur z ERP do APS jsou vstupy vkládány do *importních tabulek*. Na přiloženém CD se nachází příklad vstupních dat v podobě databáze s názvem:

```
data_example.bak
```

která se nachází v umístění:

```
Priklad_dat\Databaze_s_daty
```

Tato databáze obsahuje pouze vybranou množinu importních tabulek, které obsahují ukázková data. Databázi je nutné obnovit ze zálohy. Postup je obdobný jako při obnovení základní databáze ze zálohy, viz I.2.1.

Po obnovení databáze ze zálohy je nutné nahrát ukázková data do databáze integračního rozhraní. K tomu slouží skript s názvem:

```
import_data.sql
```

který se nachází v umístění:

```
Priklad_dat\Databaze_s_daty
```

V tomto skriptu je nutné vyplnit proměnné `@dbFrom` (databáze s ukázkovými daty, např. `data_example`) a `@dbTo` (databáze integračního rozhraní, např. `asprova_tst`). Po spuštění toho skriptu dojde k nahrání dat z databáze s ukázkovými daty do databáze integračního rozhraní. Poté je možné spustit běh integračního rozhraní.

Po dokončení běhu integračního rozhraní jsou výstupy uloženy v tabulkách a pohledech integračního rozhraní a v rámci odesílání e-mailových notifikací také v příslušných e-mailových schránkách příjemců.

Hlavní výstupy v rámci integračního rozhraní:

- *importní tabulky* (přípona `_Import`) – obsahují záznamy s importním stavem `F`, nebo `ERROR`, případně kódy chyb a příčiny chyb,
- *chybové tabulky* (přípona `_Import_Error`) – obsahují konkrétní chyby v konkrétních záznamech včetně kódů chyb a příčin chyb,
- *pohledy nad daty finálních tabulek* (přípona `_View`) – obsahují data úspěšně transformovaná do struktur systému APS,
- *logovací tabulka* (`Import_Event_Log`) – obsahuje log běhu rozhraní.

Případné záznamy, které byly archivovány v rámci běhu integračního rozhraní, jsou uloženy v příslušných archivních tabulkách.

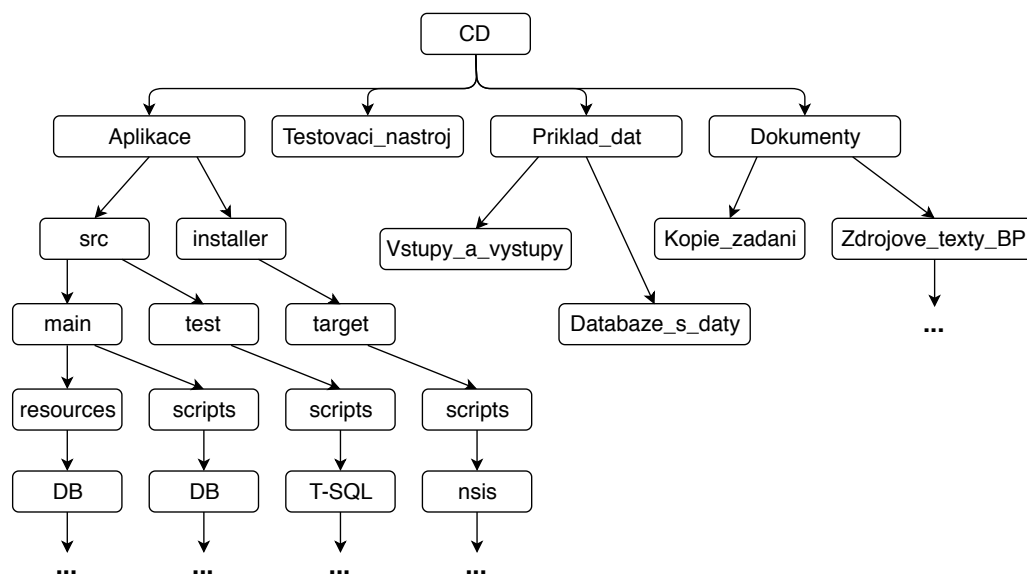
Na přiloženém CD se také nacházejí tabulky Excelu, které obsahují stejná vstupní data jako databáze ukázkových dat a k nim výsledná výstupní data. Tyto tabulky jsou uloženy v umístění:

`Příklad_dat\Vstupy_a_vystupy`.

J Obsah CD

J.1 Struktura obsahu CD

Na obrázku J.1 je uvedena základní adresářová struktura obsahu CD. V této struktuře nejsou uvedeny všechny adresáře, struktura zobrazuje pouze hlavní adresáře.



Obrázek J.1: Základní adresářová struktura obsahu CD

J.2 Popis obsahu CD

Aplikace

Tento adresář obsahuje zdrojové a binární soubory databázové aplikace, včetně instalátoru, jednotkových testů a základní databáze.

- *Zdrojové soubory aplikace:*
 - skripty, procedury, funkce, trigger, SQL joby – jsou rozděleny podle významu do adresářů (uvedeno v kapitole 6.1),
 - *umístění:* Aplikace\src\main\scripts\DB\...,

- *Zdrojové soubory jednotkových testů:*
 - jednotkové testy pro funkce a procedury – jsou rozděleny podle významu do adresářů (uvedeno v kapitole 7.1),
 - *umístění:* Aplikace\src\test\scripts\T-SQL\...,
- *Instalátor aplikace:*
 - spustitelný soubor, který nainstaluje databázové skripty do zvolené databáze,
 - *umístění:* Aplikace\installer\target\scripts\nsis,
 - *název:* ASP_Integration_2.0.0.exe,
- *Základní databáze:*
 - základní databáze pro instalaci databázové aplikace – je zabalena v archivu .zip,
 - *umístění:* Aplikace\src\main\resources\DB,
 - *název:* asp_core.zip, po extrahování asp_core.bak,
- *Konfigurační soubor pro překlad:*
 - konfigurační soubor pro vytvoření instalátoru pomocí nástroje Maven,
 - *umístění:* Aplikace,
 - *název:* pom.xml.

Testovací_nastroj

Adresář obsahuje nástroj pro testování T-SQL Test Tool. Nástroj je v podobě SQL skriptu, který vytvoří potřebnou testovací databázi a založí procedury a funkce potřebné k testování jednotkovými testy.

- *T-SQL Test Tool:*
 - *umístění:* Testovací_nastroj,
 - *název:* SetTSTDatabse.sql.

Příklad_dat

Adresář obsahuje databázi s ukázkovými vstupními daty, skript pro načtení vstupních dat do databáze integračního rozhraní a tabulky Excelu se vstupy a výstupy databázové aplikace.

- *Databáze s ukázkovými daty:*
 - databáze ve formátu `.bak` obsahující ukázková vstupní data pro databázovou aplikaci,
 - *umístění:* Příklad_dat\Databaze_s_daty,
 - *název:* data_example.bak,
- *Skript pro načtení vstupních dat:*
 - skript ve formátu `.sql` pro načtení dat z databáze `data_example.bak` do databáze integračního rozhraní,
 - *umístění:* Příklad_dat\Databaze_s_daty,
 - *název:* import_data.sql,
- *Vstupy a výstupy databázové aplikace:*
 - tabulky Excelu ve formátu `.xlsx` s ukázkou vstupních a k nim odpovídajících výstupních dat,
 - *umístění:* Příklad_dat\Vstupy_a_vystupy,
 - *názvy:* Data - <název tabulky>.xlsx.

Dokumenty

Tento adresář obsahuje dokument s bakalářskou prací, kopii zadání a zdrojové texty bakalářské práce.

- *Dokument s bakalářskou prací:*
 - dokument s bakalářskou prací ve formátu `.pdf` včetně příloh,
 - *umístění:* Dokumenty,
 - *název:* BP_Zdenek_Castoral.
- *Kopie zadání:*
 - dokumenty ve formátu `.pdf` se zadáním,
 - *umístění:* Dokumenty\Kopie_zadani,
 - *názvy:* BP_zadani_strana_1, BP_zadani_strana_2.

- *Zdrojové texty bakalářské práce:*
 - zdrojové texty a obrázky bakalářské práce potřebné k vygenerování výsledného dokumentu,
 - *umístění:* Dokumenty\Zdrojove_texty_BP\....