

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA EKONOMICKÁ

Bakalářská práce

**Sledování trendů na základě výměny elektronických
dokumentů (EDI)**

**Trending based on the exchange of electronic documents
(EDI)**

Michal Drda

Plzeň, 2016

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma

„Sledování trendů na základě výměny elektronických dokumentů (EDI)“

vypracoval samostatně pod odborným dohledem vedoucího bakalářské práce za použití pramenů uvedených v příložené bibliografii.

V Plzni, dne 23. 4. 2016

.....

Michal Drda

Poděkování

Chtěl bych poděkovat RNDr. Mikuláši Gangurovi Ph.D. za vedení mojí bakalářské práce. Dále bych chtěl poděkovat panu Ing. Lukáši Rampovi ze společnosti AIMTEC a.s. za čas, který věnoval odborným konzultacím. Chtěl bych také poděkovat svým blízkým, kteří mě při realizaci této práce podporovali.

Obsah

Úvod	7
1 Teoretická východiska	8
1.1 Informační systém	8
1.1.1 ERP	8
1.2 EDI	9
1.2.1 Hlavní přínosy EDI	12
1.3 XML	13
1.4 EDI vs. XML	15
1.5 ebXML	15
1.6 EDI zprávy a jejich formáty	16
1.6.1 Odvolávka	16
1.6.2 UN/EDIFACT	17
1.6.3 Nejvyužívanější EDI zprávy	19
1.6.4 VDA	20
1.7 Java	26
1.8 Databáze a SQL	29
2 Metodika práce	32
2.1 Představení společnosti AIMTEC a.s.	32
2.2 Vznik tématu práce	33
2.3 Výběr tématu práce	35
2.4 Zapojení ERP systémů dodavatelů a odběratele	35
2.5 Použitý software	36
2.6 Posloupnost procesů	37
2.7 Sledování odvolávek napříč dodavateli	38
3 Vlastní řešení	39
3.1 Vstupní data	39
3.2 Parser	40
3.2.1 Ošetření výskytu duplicitních zpráv	40
3.2.2 Postup čtení zprávy	40
3.2.3 Výpis extrahovaných dat	41

3.2.4	Propojení parseru s databází.....	42
3.2.5	Možnosti vylepšení parseru.....	43
3.3	Databázové operace a SQL dotazy.....	44
3.3.1	Naplnění databáze	44
3.3.2	Logika příkazů nad databází	46
3.4	Operace v programu Microsoft Excel.....	50
3.4.1	První import tabulky	51
3.4.2	Opakovaný import tabulky.....	51
3.4.3	Jak vytvořit graf	52
3.5	Trendy v logistice	53
3.5.1	Využitelnost poznatků z práce v praxi	54
	Závěr.....	56
	Seznam obrázků	57
	Seznam tabulek.....	58
	Seznam použité literatury	59
	Seznam příloh.....	62
	Přílohy	63
	Abstrakt.....	69
	Abstract.....	70

Úvod

Téma bakalářské práce vzniklo ve spolupráci s plzeňskou společností AIMTEC a.s., která vyvíjí informační systémy, zabývá se jejich implementací a poskytuje podporu a poradenství v oblasti informačních technologií. Autor si pro svoji bakalářskou práci vybral společnost AIMTEC hlavně proto, že zde v minulosti vypracoval již dvě semestrální práce, jednu z nich právě v oddělení EDI, kde se se zaměstnanci společnosti později domluvil na tématu pro tuto bakalářskou práci.

Hlavním cílem práce je identifikace trendů, které je vhodné sledovat v rámci elektronické výměny dat mezi dvěma podnikatelskými subjekty. Je důležité nebrat zde slovo trend jako statistický ukazatel, nýbrž jako „módní záležitost“. Jde tedy o informace, které zajímají management operující v logistice a výrobě.

Mezi dílčí cíle práce patří vysvětlení toho, jak EDI funguje, jaké zprávy jsou v rámci komunikace odesílány a co obsahují. Dalším dílčím cílem práce je správné zvolení typu zprávy, která bude sledována a zpracovávána, případně zpracování více typů zpráv. Cílem je také navržení vhodného mechanismu procházení a ukládání dat, která budou předmětem práce. Dílčím cílem práce je vhodná volba vizualizace zjištěných informací tak, aby posloužily k manažerskému rozhodování v reálných situacích.

Práce je rozdělena na tři hlavní kapitoly. Kapitola Teoretická východiska popisuje hlavně teoretické znalosti, potřebné k pochopení elektronické výměny dat, struktury zpráv odesílaných v rámci EDI a práce s programovacími a dotazovacími jazyky, které byly využity při práci s daty.

Kapitola pojmenovaná jako Metodika práce obsahuje představení společnosti, která poskytla datové podklady. Dále je zde popsáno, jak vzniklo téma práce, a hlavně jaký byl postup při vypracování práce. Kapitola shrnuje, jak práce s poskytnutými daty probíhala a co je výstupem této práce.

V kapitole Vlastní řešení je rozvedena praktická práce se získanými daty od jejich získání, až po vznik výstupního grafu, popsání trendů v logistice a výrobě, a využití zjištěných poznatků v praxi.

1 Teoretická východiska

1.1 Informační systém

Odborná literatura nabízí pro pojem informační systém následující definice:

- 1) Informační systém můžeme definovat jako uspořádání vztahů mezi lidmi, datovými a informačními zdroji a procedurami jejich zpracování za účelem dosažení stanovených cílů [1].
- 2) Informační systém je celek, složený z hardware a souvisejícího software, spojený s lidmi, kteří jej využívají a s procesy, které vykonávají za účelem sběru, zpracování a šíření informací, užitečných k plánování, rozhodování a kontrolování [2].

Informační systém je tedy soubor aplikací, které jsou ovládány lidmi, kteří jej využívají ke sběru a k šíření potřebných informací. Informační systém je zpravidla užívaný zaměstnanci od dělníků až po ředitele. Informace, které tyto aplikace obsahují, pomohou managementu v jeho úkolech, například v přesnosti plánování, rozhodování a kontrolování. Mluvíme zde tedy v podstatě o jádru informačního systému – o souboru aplikací ERP.

Cílem informačního systému je zpřehlednit a extrahovat procesy, které jsou v dané společnosti uskutečňovány, a s co nejvyšší efektivitou a nízkou chybovostí zamezit plýtvání zdroji. Hlavní devizou systému by měl být fakt, že management podniku i řadoví zaměstnanci budou mít přehlednější a jasné informace a instrukce k tomu, co mají dělat, tudíž se sníží vynaložené náklady a zvýší se efektivita práce. Díky těmto informacím bude mít podnik oproti ostatním firmám, které informační systém nevyužívají, konkurenční výhodu.

1.1.1 ERP

Jádrem informačního systému podniku je soubor aplikací ERP – Enterprise Resource Planning, tedy řízení zdrojů podniku. Pro uvedení zde ocitujeme opět definice:

- 1) ERP systémy představují softwarové nástroje používané k řízení podnikových dat. ERP systémy pomáhají podnikům v oblasti dodavatelského řetězce, příjmu materiálu, skladového hospodářství, přijímání objednávek od zákazníků, plánování výroby,

expedice zboží, účetnictví, řízení lidských zdrojů a v dalších podnikových funkcích [3].

- 2) ERP lze brát jako proces, pomocí kterého společnost (často výrobní podnik) řídí a integruje důležité součásti její činnosti. Informační systém ERP integruje jednotlivé moduly, jako je plánování, objednávání, skladování, obchod, marketing, finance, lidské zdroje atp [4].

Na základě těchto definic můžeme tedy říct, že jde o softwarovou část systému, která sdružuje jednotlivé moduly ERP, které mohou fungovat i samostatně. Jde například o moduly financí, lidských zdrojů, modul řízení skladu, modul plánování výroby a tak podobně.

1.2 EDI

Zkratku EDI vystihuje její anglický překlad – Electronic Data Interchange. Do češtiny ji lze volně přeložit jako Elektronická výměna dat nebo elektronický přenos dat. Pro začátek jsou opět uvedeny příklady definic z odborné literatury:

- 1) EDI – tedy elektronická výměna dat (z anglického Electronic Data Interchange) – je moderní způsob komunikace mezi dvěma nezávislými subjekty, při které dochází k výměně standardních strukturovaných obchodních a jiných dokumentů elektronickou formou [5].
- 2) EDI (Electronic Data Interchange – tj. elektronická vzájemná výměna dat) – elektronická výměna strukturovaných standardních zpráv mezi dvěma aplikacemi dvou nezávislých subjektů [6].
- 3) EDI je standardizovaná metoda k přenosu dat mezi dvěma různými počítačovými systémy nebo počítačovými sítěmi. Standardně je využita v prostředí e-commerce (elektronický obchod), jako je odesílání objednávek do skladů, sledování zásilek nebo generování faktur [7].

Druhá z definic je převzata z publikace Elektronický obchod a EDI, která vyšla již v roce 1996. Tato definice je tedy staršího data, ale je stále platná, protože ostatní dva zdroje jsou aktuální webové stránky, a definice se zásadně neliší.

V následujících několika odstavcích si proto rozebereme druhou definici do detailů a konkrétně rozepíšeme, co jednotlivá slova znamenají. Je nutno uvést, že hlavním zdrojem byla ona publikace z roku 1996, a proto bylo třeba některé teze aktualizovat, protože všechny informace by vzhledem k současnosti nebyly aktuální.

- **Elektronická výměna dat** – znamená výměnu dat elektronickou cestou, tzn. za použití elektronických přenosů „ONLINE“. Nejčastěji se v současnosti využívá připojení po internetu, https nebo ftp, dále také spojení pomocí satelitní komunikace. Dříve byly populární také sítě typu VDS Nextel, IBM IMNS, telefonní linky, případně radiové spojení. Důležité je, že výměna informací pomocí EDI je z drtivé většiny neinteraktivní – to znamená, že jsou odesílány jednotlivé zprávy obsahující kompletní blok dat a jsou odesílány najednou. Odesílání probíhá téměř vždy až po vytvoření dat, tedy ne během jejich tvorby.
- **Výměna strukturovaných dat** – strukturovaná data jsou data, která jsou definována syntaktickými pravidly. Tato pravidla vytvářejí vlastně společný jazyk pro všechny propojené aplikace. Syntaktická pravidla definují například oddělovače, délky položek, v aktuálních zprávách jsou data strukturována pomocí segmentů a jim podřazených elementů. Data jsou většinou ukládána do databáze, mají přesně definovanou délku, formát a tak dále. Struktura je definována mezinárodní normou ČSN ISO 9735 (UN/EDIFACT), od které jsou odvozené normy pro jednotlivá odvětví. Použití syntaktického jazyka a mezinárodně platného standardu (normy) je nutné pro automatické zpracování zprávy a její kompatibilitu s jinými informačními systémy. Dnes není výjimkou, že se dva ERP systémy (Enterprise resource planning – podnikový informační systém) navzájem neumí dorozumět. Potom je třeba vytvořit tzv. „mapovací konverzi“, která zajistí převod zprávy v jednom formátu na formát druhý, a oba ERP soubory aplikací se potom dokáží dorozumět.
- **Standardní zprávy** – standardní zpráva znamená, že se jedná o přesně definovaný typ zprávy, že každá položka má své místo. Příkladem z běžného života může být rozdíl mezi dopisem a formulářem. Standardní zprávou by zde byl formulář, je tedy určen standard – podobně jako ve zprávě podporované v EDI. Jak již bylo řečeno, jednotlivé standardy jsou definované normou UN/EDIFACT – United

Nation/Electronic Data Interchange for Administration Commerce and Transport. Zde jsou definovány stovky standardů pro zprávy různých odvětví, od státní správy, přes obchod, dopravu, ale například i zdravotnictví nebo stavebnictví a podobně. Kromě skupiny standardů UN/EDIFACT existuje celá řada standardů specializujících se na určité odvětví (ANSI X12, ODETTE, VDA, SEDAS). Příkladem může být standard VDA (Verband der Automobilindustrie), který se obvykle využívá pro automobilový průmysl. Již v roce 1996, kdy vyšla jmenovaná publikace, jsme se mohli dočíst, že je v plánu postupně všechny standardy nahradit standardem UN/EDIFACT, nicméně stále se hojně využívají, příkladem může být již jmenovaný standard VDA nebo standard ANSI X12, se kterým se také můžeme setkat poměrně často.

- **Mezi dvěma aplikacemi** – EDI funguje při výměně dat mezi dvěma aplikacemi, na rozdíl od aplikací, které jsou stavěny jako jednoúčelové (homebanking – aplikace určená ke komunikaci zákazníka s bankou). V případě EDI předává jedna aplikace jiné aplikaci data zcela automaticky a bez nutnosti nebo s minimální nutností lidských zásahů. Během zavádění se sice využívají zjednodušené aplikace, které mají jednoduché uživatelské rozhraní pro ruční vkládání, cílem je ale komunikace aplikace s aplikací, tedy proces plně automatický.
- **Mezi dvěma nezávislými subjekty** – z hlediska návrhu pro EDI systémy je nutné definovat, zda se jedná o závislé nebo nezávislé subjekty. Pokud má firma dvoje kanceláře, každou na jiném konci města, nejedná se o nezávislé subjekty, ale o jeden podnik. V tomto případě je možné definovat jednodušší strukturu, než při zavedení EDI. Nároky na bezpečnost jsou nesrovnatelně nižší než u nezávislých subjektů, právní problematiku a audit není třeba při vzájemné komunikaci vůbec řešit. Nemá tedy žádný smysl EDI do podniku pro vnitropodnikovou komunikaci zavádět. Pokud však kupříkladu výhradní odběratel zavádí EDI, je sice pro jeho dodavatele nezbytně nutné se napojit, ale z koncepčního a především legislativního hlediska se jedná o nezávislé subjekty, a právní stránka zde již hraje významnou roli. Projekt je nutné řešit jako EDI projekt. Výhradní odběratel sice může vydefinovat vlastní datové struktury, zabezpečení i komunikaci, ale je pro něj výhodnější se držet standardů, vzhledem k tomu, že v budoucnu stejně bude muset komunikovat s jiným (větším)

subjektem – výhradní odběratel se státní správou, státní správa s Evropskou Unií atd., a bude nucen používat jeho standardů. Je pochopitelně všestranně výhodnější vybudovat kvalitní systém, kde budou všichni účastníci užívat stejných standardů, než později muset měnit systém celý. V dnešní době lze řešit i situaci, kdy formáty obou partnerů nejsou stejné, je však nesrovnatelně složitější komunikaci zprůchodnit. Je třeba tvorby složitých konverzí a změn mapování, je tedy nadále výhodnější využívat mezinárodních standardů.

EDI tedy můžeme také definovat jako výměnu zpráv, které mají stejnou formu a jsou psány stejným „jazykem“, mezi aplikacemi dvou nezávislých partnerů [6].

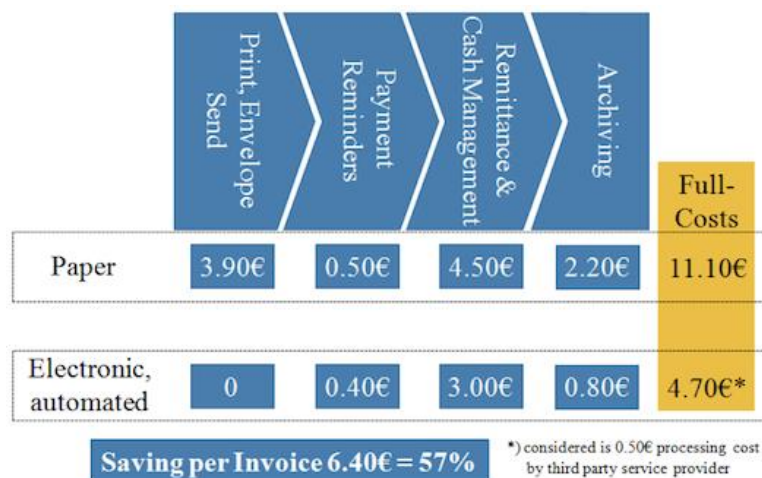
1.2.1 Hlavní přínosy EDI

Těmi stěžejními přínosy, které definuje již zmiňovaná kniha Elektronický obchod a EDI, je bezesporu úspora času, peněz a obecně i materiálu, z toho plynoucí důsledky, například důsledky ekologické.

Pokud se zaměříme na úsporu finančních prostředků plynoucí ze zavedení e-invoicingu (elektronické fakturace), tedy potažmo zavedení EDI, existují mnohé studie, které dokazují, že díky elektronické výměně dat lze dosáhnout vysoké úspory s rychlou návratností (jeden až jeden a půl roku).

Příkladem může být studie společnosti Billentis, která říká, že odesílatel faktury může za předpokladu, že využije elektronickou cestu – e-invoicing, EDI, ušetřit v průměru 57% nákladů. Do uvažovaných nákladů jsou započítány náklady na vytvoření, vytištění a odeslání, na kontakt s příjemcem faktury a připomenutí a převod peněz na účet [8].

Příjemce má podle studie ušetřit až 62% nákladů, a jeho náklady plynou především z přijímání, validace a potvrzování, dále z komunikace uvnitř firmy, komunikace s bankou a archivování. O výsledcích informuje článek, z něhož jsou v práci vytaženy následující dva obrázky shrnující předchozí údaje [8].



Obrázek 1: Úspora plynoucí z elektronizace fakturace ze strany odesílatele. Zdroj: [8].



Obrázek 2: Úspora plynoucí z elektronizace fakturace ze strany příjemce. Zdroj: [8].

1.3 XML

Pojem XML (eXtensible Markup Language) je v dnešní době již poměrně známý. Jednoduše řečeno, jde o „značkovací jazyk“, který umožňuje „obalit“ informace takzvanými (tagy) a tak je umístit do určité skupiny. Hlavní přínos spočívá v tom, že jednotlivé XML tagy si může uživatel jazyka definovat sám. Jako příklad lze uvést reprezentaci ceníku [9], příklad, který lze najít v publikaci Jiřího Koska.

```

<ceník>
  <výrobek>
    <název>SuperInkJet 120TDi</název>
    <popis>Moderní inkoustová tiskárna</popis>
    <cena>12.000,-</cena>
  </výrobek>
  <výrobek>
    <název>OfficeCom 56K</název>
    <popis>Faxmodem s hlasovými funkcemi</popis>
    <cena>3.500,-</cena>
  </výrobek>
  <výrobek>
    <název>CD-RW 3246</název>
    <popis>Mechanika CD-ROM s možností vypalování a přepisu</popis>
    <cena>9.500,-</cena>
  </výrobek>
</ceník>

```

Obrázek 3: Cenník vytvořený za pomoci XML kódu. Zdroj: [9].

Na první pohled vidíme, že v takto strukturovaných datech se bude vyhledávat mnohem snáze, než v datech, která tagy opatřena nejsou. Díky tomu má XML uplatnění jak ve webových stránkách, tak například právě ve zmiňovaném EDI, nebo v podstatě v jakékoliv elektronické komunikaci. Hlavní důvod, proč je důležité jazyk XML v práci uvést, je ten, že struktura XML je velmi blízká EDI. XML se využívá k zajištění čitelnosti převáděných EDI zpráv a velmi časté je například převedení zpráv ve standardu UN/EDIFACT do jednoduchého XML, které je sice velikostně mnohem náročnější, ale jednoznačně se v něm lépe orientuje a obecně je mnohem čitelnější.

Dnes je v módě zkratka B2B. V praxi se však nejedná o nic jiného, než o výměnu informací mezi obchodními partnery v elektronickém formátu. Jako vhodný formát pro přenos dat se jeví právě formát XML, který je velice jednoduchý a podporuje ho mnoho počítačových platforem. Pomocí XML si firmy mohou vyměňovat objednávky, faktury a mnoho dalších údajů [9].

Hlavní výhodou XML oproti EDI je podle autora citované publikace fakt, že XML je oproti EDI mnohem jednodušší na implementaci. Datové formáty EDI bývají velmi složité a častým problémem je také nekompatibilita jednotlivých systémů, kde je nutné tvořit mapovací konverze, a soubory při cestě mezi dvěma systémy překládat do potřebných cílových formátů mnohem složitěji, než to lze za použití jednoduchého XML [9].

Předchozí odstavec je úryvkem z již citované publikace Jiřího Koska. Můžeme zde najít blízkou podobnost právě se systémy EDI, o kterých je v publikaci následující odstavec. Ve shrnutí je v něm řečeno, že XML je mnohem jednodušší na implementaci než EDI, a že datové formáty EDI bývají velmi složité.

1.4 EDI vs. XML

Pokud jde o budoucnost, očekává se, že v oblasti elektronické výměny dat poroste podíl XML transakcí nad klasickými EDI transakcemi, nicméně oba přístupy mají své pro a proti.

- Jedním z hlavních kladů XML, oproti EDI, je již uvedená čitelnost dat.
- Často se uvádí, že EDI je dražší a náročnější na implementaci. Obecně tento fakt platí v případě, že všechny konverze a procesy nastavuje firma poprvé. Je však třeba brát v potaz, že v EDI bylo již prakticky vše vymyšleno, a tudíž je mnohem rychlejší EDI implementovat, než implementovat XML.
- U XML se postupně zvyšují nároky na diskovou kapacitu, zprávy bývají obvykle datově náročnější, než zprávy ve formátech EDI.
- EDI je zavedená dlouho trvající služba, je tedy možné jí věřit, je to pro firmy menší riziko, než nové cesty pomocí XML.
- XML je nová technologie, která má potenciál být v budoucnu levnější a celkově výhodnější, než klasické EDI.
- EDI je oproti XML mnohem robustnější řešení. Výhodou XML naopak může být jeho flexibilita.

[10]

1.5 ebXML

Když bylo zmíněno EDI a XML, je nutností znít alespoň několik vět o takzvaném ebXML. Lze najít několik různých definic toho, co to ebXML je, nejlepší možnou definicí by však měla být definice z oficiálního webu sloužícího jako zdroj informací o ebXML.

ebXML (Electronic Business using eXtensible Markup Language) je modulární sada specifikací, které umožňují podnikům všech velikostí v jakékoliv zeměpisné poloze podnikat

pomocí internetu. Pomocí ebXML mohou společnosti standardizovat jejich elektronickou komunikaci, řídit obchodní vztahy, odesílat a přijímat data a definovat obchodní procesy [11].

Pokud svými slovy rozvedeme, co to ebXML je, dospějeme nejspíš k závěru, že jde o standardizovaný XML formát, který se využívá k B2B komunikaci. Jde tedy o značkovací jazyk, který obsahuje data ve standardizovaných – centrálně definovaných segmentech (značkách), a v přesně definovaném formátu. Takto bylo plánováno využití v době, kdy tento standard pod záštitou světového centra pro elektronický obchod UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) a organizace pro rozvoj strukturovaných informačních standardů OASIS (Organization for the Advancement of Structured Information Standards) vznikal [11].

Realita je však v dnešní době odlišná. Vzhledem k rozdílnému způsobu využití se pro elektronickou výměnu dat používá vlastních XML formátů nebo již zavedených EDI standardů, takže ebXML není tak používané, jak se zpočátku čekalo.

1.6 EDI zprávy a jejich formáty

Ve většině případů je EDI komunikace přeložena do jednoduchých XML formátů, které jsou čitelnější, a lépe se v nich orientuje. Je však podstatné zabývat se i surovým formátem EDI zpráv. Proto v této kapitole bude rozebrána struktura zprávy VDA4905. Z názvu je již patrné, že se jedná o formát VDA (v originálu Verband der Automobilindustrie), jenž můžeme z němčiny přeložit jako formát sdružení automobilového průmyslu. Vzhledem k častému využití formátu UN/EDIFACT bude v této kapitole rozebrán také protějšek zprávy VDA4905 právě ve formátu UN/EDIFACT, tedy zpráva DELFOR. Oba tyto příklady by měly reprezentovat zprávu zvanou „odvolávka“, v angličtině nazývanou „call-off“.

1.6.1 Odvolávka

Před rozepsáním jednotlivých částí zprávy a jejich celkovým strukturálním popisem bude určitě vhodné popsat, co to vlastně „odvolávka“ je. Jednoduše řečeno, je to zpráva, kterou odesílá spotřebitel (odběratel) materiálu/komponent, svému dodavateli, aby dodavatel věděl, jaký materiál a v jakém množství bude na cílový termín dodání odběratel potřebovat doručit.

Dodavatel se podle těchto předpovědí – odvolávek může dopředu připravit a naplánovat tak svoji výrobu, aby byl schopen dostát požadavkům odběratele (zpravidla velká společnost, typickým příkladem může být automobilka).

Základní pravidla, kterými se odvolávka, jako typ zprávy řídí, můžeme stručně shrnout do několika následujících bodů:

- Odvolávka je zpráva, která obsahuje předpovědi požadovaného množství daného materiálu, vztažené k určitému datu nebo časovému horizontu.
- V rámci jedné zprávy může dodavatel odeslat předpověď na více druhů materiálu, které dodává jeden konkrétní dodavatel.
- V rámci jedné zprávy jsou stanoveny požadavky na více časových úseků nebo na konkrétní dny.
- Odvolávkám předchází sjednání kontraktu nebo závazná objednávka.
- Důležitý fakt je, že pro daný materiál od daného dodavatele na daný kontrakt je aktuální vždy jen poslední informace – poslední zpráva/odvolávka. Znamená to tedy, že nové zprávy v podstatě „překrývají“ ty starší.

1.6.2 UN/EDIFACT

Velké množství EDI komunikace v současné době probíhá v rámci zavedeného standardu UN/EDIFACT, proto v práci alespoň zevrubně rozebereme strukturu, jakou zprávy v tomto formátu mají.

Hlavní devizou formátu UN/EDIFACT je, že oproti zmiňovanému formátu VDA je mnohem lépe čitelný, aniž by musel být převeden do XML formátu. Pro následné představení bude využita zpráva DELFOR ve verzi D97A, standardu EDIFACT. Zpráva se skládá z takzvaných segmentů, které mají přesně definované názvy a jsou vždy na řádku samostatně. Uvnitř jednotlivých segmentů jsou použity prvky první úrovně a prvky druhé úrovně, které již nesou danou konkrétní informaci a je jich obvykle v jednom řádku více.

Přesně definované jsou také oddělovače a informativní znaky. Apostrof „‘“ slouží k ukončení segmentu, měl by být na řádce vždy posledním znakem. Znak plus „+“ slouží jako oddělovač jednotlivých elementů na nejvyšší úrovni, je ve zprávě vypsán i v případě, že

elementy, které odděluje, nejsou využity. Za posledním elementem uvažovaným v řádku již není třeba vkládat znak plus. K oddělení jednotlivých dat zanořených v elementu (elementy druhé úrovně) se využívá jako oddělovač dvojtečka „:“. Pokud je potřeba uvolnit znak, který má určitou funkci v syntaxi formátu, například znak plus, je před něj vepsán znak otazník „?“ , funguje zde tedy jako takzvaný uvolňovač. Dále je možné využít znaku mezery „ “ k rezervování pozice, případně znaku tečky „.“ v případě použití desetinného čísla [12].

Pro představu, jak oddělovače ve zprávě fungují, je zde uvedeno několik příkladů, za použití elementů první úrovně:

SEG+A+B+C+D'	Všechny čtyři elementy jsou obsaženy.
SEG+A++C+D'	Jsou obsaženy všechny elementy kromě B.
SEG+A'	Pouze element A je obsažen v řetězci.
SEG+A+++'	Druhá možnost, zápisu pro samotné A.
SEG++++D'	V řetězci je pouze element D.

Tabulka 1: Popis elementů první úrovně ve standardu UN/EDIFACT. Zdroj: [13].

Analogicky fungují oddělovače i pro druhou úroveň elementů, v následujících příkladech bude představena jejich funkcionalita:

SEG+A+B1:B2:B3:B4+C+D'	Všechny elementy obou řádů obsaženy.
SEG+A+:B2:B3:B4+C+D'	Chybí element B1 druhého řádu.
SEG+A+B1:B2::B4+C+D'	Chybí element B3 druhého řádu.
SEG+A+B1:B2+C+D'	Chybí element B3 a B4 druhého řádu.
SEG+A++C+D'	Chybí elementy B1 až B4 druhého řádu.

Tabulka 2: Popis elementů druhé úrovně ve standardu UN/EDIFACT. Zdroj: [13].

Po popsání základních pravidel je prostor k popisu jednotlivých segmentů, aby měl čtenář přehled o tom, co se ve zprávě DELFOR dle normy UN/EDIFACT vyskytuje za informace. V tabulce (příloha A) o čtyřech sloupcích bude uveden výčet jednotlivých segmentů, jejich význam ve zprávě a údaj, zda je segment ve zprávě nutný („M“ - Must), či nikoliv („C“ – Conditional), a také zanořenost segmentu nebo skupiny segmentů. Segmenty tedy budou rozděleny do nadřazených skupin. Je podstatné uvést, že text vychází z dat uváděných pro verzi zprávy D97A, struktura zprávy se může dle podniku mírně lišit, některé segmenty se vůbec plnit nemusí, jiné ano, některé segmenty mohou být ve zprávě zopakovány vícekrát, toto platí především pro jednotlivá odvolávaná množství a data. Podnik také může využívat

jednotlivé segmenty k odesílání různých informací, proto nemusí popis segmentů bez výjimky platit vždy tak, jak je znázorněný tabulkou v příloze A.

1.6.3 Nejvyužívanější EDI zprávy

Pochopitelně existuje mnoho typů zpráv, které mohou být reprezentovány různými standardy. V rámci představy o existenci alespoň nějakých typů zpráv je možné se zaměřit například na formát EDIFACT, který je v současnosti asi nejrozšířenější. Zpráv, které mohou být přenášeny pomocí formátu EDIFACT, je velké množství, bylo by však velmi náročné vměstnat jejich podrobný popis do jedné bakalářské práce. Jako názorná ukázka mohou být jmenovány například následující zprávy se stručným popisem:

- DESADV (Despatch advice – avízo k odeslání zboží/materiálu) – nese informaci o dodávaném zboží nebo materiálu. Tato zpráva bývá odesílána před fyzickou dodávkou zboží nebo materiálu, aby bylo umožněno odběrateli připravit si dostatek kapacity na příjem a naskladnění dané zásilky [14].
- ORDERS (Purchase order – objednávka) – jde o zprávu, kterou odesílá objednatel zboží jeho dodavateli. Zpráva obsahuje informace o specifikaci zboží a informace o místu, případně času dodání [15].
- ORDRSP (Order response – potvrzení objednávky) – tato zpráva je potvrzením objednávky, jde o reakci na přijatou objednávku. Tuto zprávu odesílá dodavatel svému odběrateli a odesílá v ní informace o materiálu/zboží, který je schopen dodat, na jaké místo a kdy. Jde v podstatě o potvrzení objednávky a upřesnění možností dodavatele [16].
- INVOIC (Invoice – faktura) – zpráva je zpravidla odesílána dodavatelem jeho odběrateli jako reakce a požadavek na zaplacení jeho dodávky. Faktura může obsahovat údaje z více transakcí, údaje o dopravě, platebních podmínkách, celní a daňové odvody a podobně. Faktura by měla obsahovat všechny informace, které obsahuje faktura papírová, včetně podpisu [17].
- INVRPT (Inventory report – přehled o skladových zásobách) – zpráva obsahuje současný nebo budoucí stav skladu a může být odesílána jak dodavatelem, tak odběratelem [18].

- APERAK (Application error and acknowledgement message – zpráva o zpracování přijaté zprávy) – tato zpráva obsahuje informace o přijetí nebo o chybě při přijímání zpráv. Je odesílána příjemcem zpráv jejich odesílateli [19].

[20]

1.6.4 VDA

Zprávy ve formátu VDA, tedy zprávy primárně určené k použití v automobilovém průmyslu mají na rozdíl od rozšířenějších UN/EDIFACT zpráv velmi odlišný a specifický formát. Většina informací je ve zprávě zakódována pomocí řetězců o přesně definované délce a oddělovače jako v EDIFACTU zde nejsou. V bakalářské práci se pracuje se zprávami typu VDA4905/1, které jsou odpovídajícím ekvivalentem typu zprávy DELFOR v rámci EDIFACT skupiny formátů. Podobně jako v předchozí podkapitole bude tedy rozebrána struktura odvolávky ve formátu VDA a následně uvedeny příklady dalších zpráv, které se v rámci formátu VDA posílají.

Podobně jako u zprávy DELFOR v předchozí podkapitole, i zpráva VDA4905/1 je složená ze segmentů a jejich elementů. Ve formátu VDA však nejsou segmenty definovány zkratkami tří písmen, ale zkratkou tří číslic. Konkrétně u odvolávky VDA4905/1 jsou to trojčíselná čísla začínající pětkou, následuje jednička a potom číslice od jedné do devíti. Přesný popis jednotlivých segmentů obsahuje tabulka číslo 3. Tabulka obsahující řádky s informacemi o jaký segment se jedná, jeho popis a informaci o nutnosti výskytu. „M“ zde znamená must – nutnost výskytu, „C“ zde znamená conditional – možnost výskytu.

Seg.	Popis	M/C
511	Hlavička zprávy obsahující instrukce o dodávce, ve zprávě je pouze 1x	M
512	Unikátní data o obchodních partnerech, materiálu a vykládce, 1x/cyklus, předchůdci: „511“ nebo „513“ / „514“ / „517“ / „518“	M
513	Odvolávková data, obsahuje konkrétní předpovědi, 1x/cyklus, předchůdce: „512“	M
514	Další předpovědi, 1x/cyklus, předchůdci: „513“ / „514“	C
515	Doplňující údaje k odvolávce, 1x/cyklus, předchůdce: „514“	C
517	Balící instrukce, i několikrát/cyklus, předchůdci: „513“ / „514“ / „515“ / „517“, pouze pokud daná regulace balení	C

518	Text, i několikrát/cyklus, předchůdci: „513“ / „514“ / „515“ / „517“ / „518“, pouze v nutnosti	C
519	Patička odvolávky, ve zprávě pouze 1x, předchůdci: „513“ / „514“ / „515“ / „517“ / „518“	M

Tabulka 3: Zpráva VDA 4905 – stručný popis jednotlivých segmentů. Zdroj: Vlastní zpracování, [21].

Za hlavičkami segmentů jsou příslušné elementy, které nesou informaci ať už udanou kódem pro danou událost, nebo jsou zadána uživatelem – v systému odběratele. Elementy mají vždy jasně stanovenou délku řetězce. V případě, že je v elementu alfa-numerický řetězec méně znaků, než je délka elementu, volné pozice jsou vyplněny mezerami. V případě, že se jedná o numerický formát, potom jsou na volných pozicích takzvané „leading zeros“, tedy od počátku jsou nevyužité pozice vyplněny nulami. Toto platí i v případě, že je pole zcela nevyužito. Nedochozí tedy ke zkrácení řetězce, jelikož by se tak narušila struktura zprávy.

Všechna data ve formátu datum jsou vložena ve formátu YYMMDD, kde YY znamená rok, MM znamená měsíc a DD znamená den. Pokud tedy chceme zapsat datum 1. únor 2016, řetězec bude vypadat takto: 160201. Ve zprávách se vyskytují také jiná data na polích, kde se dá očekávat datum. Většinou reprezentují určitý kód, znamenající nějakou specifickou situaci, nebo predikci ve formátu týdnů. O těchto výjimkách pojednává jedna z podkapitol.

V následujících podkapitolách budou rozebrány detailně jednotlivé segmenty, které nesou důležité informace potřebné v rámci sledování přesnosti predikovaných množství požadovaného materiálu. V prvním sloupci je opět informace, zda pole musí („M“ – must) být vyplněno, či nikoliv („C“ – conditional), ve druhém sloupci je informace, zda se jedná o „A“ alfanumerický nebo „N“ numerický řetězec, ve třetím sloupci je délka řetězce, ve čtvrtém jeho pozice v řádce a v posledním sloupci je popis pole.

1.6.3.1 Segment 511

Segment s číslem „511“ obsahuje údaje o příjemci a odesílateli, dále kódy a přenosová čísla, datum přepravy a další údaje. Přesný popis segmentu lze nalézt v tabulce číslo 4.

C/M	A/N	Délka	Od-do	Popis
M	N	3	1-3	Konstanta značící typ segmentu, „511“
M	N	2	4-5	Verze, „02“

M	N	9	6-14	Kód odběratele - příjemce materiálu, přiděleno dodavatelem
M	A	9	15-23	Kód dodavatele - odesílatel materiálu, přiděleno odběratelem
M	A	5	24-28	Kód předchozího přenosového čísla
M	N	5	29-33	Kód nového přenosového čísla
M	N	6	34-39	Datum přenosu zprávy
C	N	6	40-45	Počáteční datum kumulativního příjmu materiálu
M	A	83	46-128	Vyplněno mezerami

Tabulka 4: Zpráva VDA 4905 – odvolávka, popis segmentu „511“. Zdroj: Vlastní zpracování, [21].

1.6.3.2 Segment 512

Segment s číslem „512“ obsahuje konkrétní údaje o místě skladu, informaci o identifikaci materiálu, identifikaci dodavatele i odběratele a další potřebné údaje o dodávce. Celý popis obsahuje tabulka číslo 5.

C/M	A/N	Délka	Od-do	Popis
M	N	3	1-3	Konstanta značící typ segmentu, „512“
M	N	2	4-5	Verze, „01“
M	A	3	6-8	Kód závodu zákazníka
M	N	9	9-17	Číslo odvolávky, počítadlo pro každý materiál a závod zvlášť
M	N	6	18-23	Datum odvolávky
M	N	9	24-32	Číslo předchozí odvolávky, analogicky jako s číslem odvolávky
M	A	22	39-60	Kód - ID zákazníka - odběratele (příjemce)
C	A	22	61-82	Kód - ID dodavatele
C	N	12	83-94	Číslo objednávky - stanoveno obvykle na jeden rok
M	A	5	95-99	Identifikace brány/místa na skladu
M	A	4	100-103	Identifikace odpovědné osoby zákazníka
M	A	2	104-105	Měrná jednotka
M	A	1	106	Konstanta „L“, značí četnost dodávek
C	N	1	107	Kolik měsíců zbývá do uvolnění do výroby
C	N	1	108	Kolik měsíců zbývá do uvolnění materiálu
M	A	1	109	„S“ pro sériový a „E“ pro oddělený materiál
C	A	7	110-116	Doplňková data o zákazníkovi
C	A	7	117-123	Lokace skladu
M	A	5	124-128	Vyplněno mezerami

Tabulka 5: Zpráva VDA 4905 – odvolávka, popis segmentu „512“. Zdroj: Vlastní zpracování, [21].

1.6.3.3 Segment 513

Segment „513“ je stěžejním segmentem odvolávky. Obsahuje informace o datech a množstvích, kdy je požadováno dodat materiál odběrateli. Celý popis segmentu s délkou každého řetězce, nutností existence údajů ve zprávě a délkách řetězců lze najít v tabulce 6.

C/M	A/N	Délka	Od-do	Popis
M	N	3	1-3	Konstanta značící typ segmentu, „513“
M	N	2	4-5	Verze, „01“
M	N	6	6-11	Datum poslední přijatého dokladu na daný materiál a závod
M	N	8	12-19	ID posledního přijatého dokladu na daný materiál a závod
M	N	6	20-25	Datum posledního dokladu při odeslání
M	N	12	26-37	Množství na posledním zaznamenaném dokladu (3 Desetinná místa)
M	N	10	38-47	Kumulativní množství, které bylo dosud zaznamenáno
M	N	6	48-53	Datum očekávaného přijetí materiálu, 1.
M	N	9	54-62	Množství očekávaného přijetí materiálu, 1.
C	N	6	63-68	Datum očekávaného přijetí materiálu, 2.
C	N	9	69-77	Množství očekávaného přijetí materiálu, 2.
C	N	6	78-83	Datum očekávaného přijetí materiálu, 3.
C	N	9	84-92	Množství očekávaného přijetí materiálu, 3.
C	N	6	93-98	Datum očekávaného přijetí materiálu, 4.
C	N	9	99-107	Množství očekávaného přijetí materiálu, 4.
C	N	6	108-113	Datum očekávaného přijetí materiálu, 5.
C	N	9	114-122	Množství očekávaného přijetí materiálu, 5.
M	A	6	123-128	Vyplněno mezerami

Tabulka 6: Zpráva VDA 4905 – odvolávka, popis segmentu „513“. Zdroj: Vlastní zpracování, [21].

1.6.3.4 Segment 514

Segment „514“ je jakýmsi pokračováním segmentu „513“. Jeho existence není nutná, v případě, že data o požadovaném doručení materiálu pojme segment „513“. V případě, že je dat více, je využito až několik segmentů „514“. Celkový popis lze najít v tabulce číslo 7.

C/M	A/N	Délka	Od-do	Popis
M	N	3	1-3	Konstanta značící typ segmentu, „514“
M	N	2	4-5	Verze, „01“
M	N	6	6-11	Datum očekávaného přijetí materiálu, pokračuje z „513“

M	N	9	12-20	Množství očekávaného přijetí materiálu, pokračuje z „513“
C	N	6	21-26	Datum
C	N	9	27-35	Množství
C	N	6	36-41	Datum
C	N	9	42-50	Množství
C	N	6	51-56	Datum
C	N	9	57-65	Množství
C	N	6	66-71	Datum
C	N	9	72-80	Množství
C	N	6	81-86	Datum
C	N	9	87-95	Množství
C	N	6	96-101	Datum
C	N	9	102-110	Množství
C	N	6	111-116	Datum
C	N	9	117-125	Množství
C	N	3	126-128	Vyplněno mezerami

Tabulka 7: Zpráva VDA 4905 – odvolávka, popis segmentu „514“. Zdroj: Vlastní zpracování, [21].

1.6.3.5 Segment 519

Segment číslo „519“ je patičkou celé zprávy, která obsahuje segmenty s číselným označením od „511“ do „519“. Všechny segmenty z důvodu nepotřeby zmíněny nebyly, nicméně patička obsahuje součty jednotlivých segmentů a informace v ní jsou užitečné pro případnou kontrolu zprávy. Celkový popis struktury segmentu lze najít v tabulce číslo 8.

C/M	A/N	Délka	Od-do	Popis
M	N	3	1-3	Konstanta značící typ segmentu, „519“
M	N	2	4-5	Verze, „02“
M	N	7	6-12	Počet přenesených segmentů „511“
M	N	7	13-19	Počet přenesených segmentů „512“
M	N	7	20-26	Počet přenesených segmentů „513“
M	N	7	27-33	Počet přenesených segmentů „514“
M	N	7	34-40	Počet přenesených segmentů „517“
M	N	7	41-47	Počet přenesených segmentů „518“
M	N	7	48-54	Počet přenesených segmentů „519“
M	N	7	55-61	Počet přenesených segmentů „515“
M	A	67	62-128	Vyplněno mezerami

Tabulka 8: Zpráva VDA 4905 – odvolávka, popis segmentu „519“. Zdroj: Vlastní zpracování, [21].

Za zmínku stojí také fakt, že v předchozí tabulce jsou sčítány i segmenty s označením „511“ a „519“. Segment „519“ slouží v praxi jako počítadlo segmentů a nejen kvůli informaci pro uživatele, který se zprávou může přijít do styku, ale také proto, aby byla zajištěna výstupní kontrola obsahu zprávy, tedy že zpráva obsahuje správný počet segmentů daného typu, včetně hlavičky a patičky.

1.6.3.6 Speciální kódová označení na poli datum

V následující tabulce jsou obsažena kódová označení, která lze nalézt na pozici data vztaženého k požadované dodávce v rámci segmentu „513“ nebo segmentu „514“. Jde o specifické příklady, které značí určitou událost. Speciální kódová označení popisuje tabulka číslo 9.

Kód	Popis
000000	Již neexistuje další záznam - konec segmentu
222222	Pro daný materiál a závod není aktuálně požadavek
333333	Materiál, který byl odvoláván a nebyl dodán - zpožděná dodávka
444444	Materiál je požadován pokud možno ihned, co nejdříve
555555	Všechna množství a datумы po tomto kódu jsou ve formátu týdnů či měsíců
999999	Výhled množství materiálu na několik měsíců

Tabulka 9: zpráva VDA 4905 – odvolávka, kódová označení. Zdroj: Vlastní zpracování, [21].

Je vhodné zmínit, že kód 555555 se vyskytuje v drtivé většině zpráv. Využívá se k dlouhodobým predikcím, kdy je vystaven požadavek na dodání materiálu na určitý týden nebo na určitý interval daný rozpětím týdnů.

Aby si čtenář dovedl představit, co se skrývá ve zprávě na polích určených pro datum, jsou základní zákonitosti vysvětleny následujícím bodovým seznamem:

- V případě požadavku na určitý týden je formát data ve tvaru YY00WW, kde YY znamená rok a WW daný týden, jako příklad lze uvést řetězec 160045, tedy týden číslo 45 roku 2016, jemuž odpovídá datum 7. 11. 2016 až 13. 11. 2016.
- V případě požadavku na časový interval daný rozpětím týdnů bude formát data ve tvaru YYWWWW, kde opět YY znamená rok a WW týden. Jako příklad lze uvést řetězec 160104. Řetězec obsahuje informaci o tom, že odběratel požaduje dané množství

materiálu na první až čtvrtý týden roku 2016 včetně, tedy interval od 4. 1. 2016 do 31. 1. 2016.

- V případě požadavku na určitý měsíc by byl formát data ve tvaru `YYMM00`, kde `YY` opět značí rok a `MM` znamená daný měsíc, na který je požadavek vystaven. Příkladem může být řetězec `160300`, tedy požadavek na dodání materiálu v rámci měsíce března roku 2016, tedy požadavek na doručení daného množství materiálu v průběhu 1. 3. 2016 až 31. 3. 2016.

[21]

1.7 Java

Hlavní metoda, která je vyvolána po spuštění programu jako první, se musí „jmenovat“ `main` a musí být v programu vždy uvedena. Tato metoda musí být „zapouzdřena“ (tj. musí být uvnitř) v nějaké třídě. Třída by se měla jmenovat stejně jako název souboru, ve kterém je napsána, měla by být označena jako `public`. Metoda `main` musí mít přesně tuto podobu:

```
1 public class ExampleCode{
2
3     public static void main(String []args){
4         //Tělo metody main.
5     }
6 }
```

Obrázek 4: Deklarace třídy v jazyku Java. Zdroj: Vlastní zpracování.

Pokud metoda `main` nespĺňuje tyto požadavky, není možné program přeložit ani spustit. Je tedy potřeba tato pravidla dodržovat [22].

Základní datové typy (též primitivní datové typy – primitive data types) jsou v Javě několika druhů – celočíselné, znakové, logické, reálné a prázdný datový typ `void`, který se používá jen u metod. Java u každého datového typu definuje jeho velikost, tj. počet bajtů, které zabírá v paměti [22].

V tabulce číslo 10 jsou obsaženy datové typy. V tabulce je uveden výčet nejpoužívanějších datových typů a jejich obvyklé použití.

Typ	Popis
<code>int</code>	celočíselná čísla s rozsahem až +- 2 147 483 648

long	celočíslná čísla s rozsahem až +- 9 233 372 036 854 775 808
float	reálná čísla, mohou obsahovat znaménka i exponenty
double	reálná čísla, mohou obsahovat znaménka i exponenty, řádově vyšší rozsah oproti float
boolean	binární hodnota, obsahuje buď true nebo false
String	alfanumerický řetězec, může obsahovat jakékoliv znaky
char	Jeden znak

Tabulka 10: Datové typy v programovacím jazyku Java. Zdroj: Vlastní zpracování, [22].

Často se využívá příkazů `if`, `if-else`, `for`, `while` a `do-while`. V následujících několika řádcích bude shrnuta základní teorie pro tyto příkazy.

Příkaz `if` je jeden z nejvyužívanějších příkazů a Java jej umožňuje použít jak v neúplné podmínce – pouze `if`, tak i v případě `if-else` [22].

Př.: Berme v úvahu, že před následujícím kódem existuje definice barvy, definice třídy a metody `main`. Tato barva může být definována buď jako `black` (černá), `white` (bílá), nebo jako `gray` (šedivá). Kód podle toho, jak je barva definována vypíše text s informací, o jakou barvu se jedná.

```

9 ▾
10
11
12 ▾
13
14
15 ▾
16
17
if (barva == black) {
    System.out.println("Barva je černá");
}
else if (barva == white) {
    System.out.println("Barva je bílá");
}
else {
    System.out.println("Barva je šedivá");
}

```

Obrázek 5: Podmínka `if`, `else if`, `else`. Zdroj: Vlastní zpracování.

Příkaz `for` probíhá tak, že se na počátku vyhodnotí počáteční výraz (výraz.start), otestuje se, zda je splněna podmínka (výraz.stop) jako pravdivá – `true`, provede se příkaz uvnitř cyklu, na závěr se provede iterace (výraz.iter). Pak začíná další obrátka cyklu, nyní již rovnou testováním podmínky (výraz.stop) [22].

Př.: Před výňatkem z kódu musí samozřejmě existovat definice třídy, musí zde existovat metoda `main`. Výstupem kódu bude pět řádků, každý bude obsahovat číslo, postupně od 0 do 4. Protože cyklus bude probíhat, dokud bude `i` rovné číslu menšímu než 5, a pokud bude tato podmínka splněna, vypíše se `i` a poté se `i` navýší o 1 a cyklus se zopakuje.

```

20 -   for (int i=0;i<5;i++) {
21     System.out.println(i);
22   }

```

Obrázek 6: Cyklus for. Zdroj: Vlastní zpracování.

Příkaz `while` testuje podmínku cyklu před průchodem cyklem. To znamená, že cyklus nemusí proběhnout ani jednou. Používáme jej v případě, že závisí ukončovací podmínka na nějakém příkazu v těle cyklu [22].

Př.: Jako u předchozích příkladů, někde nad cyklem-příkazem `while` by měla být definována třída, metoda `main` a v tomto případě také proměnná `x`, která je typu `int`. V případě, že je proměnná větší než 9, tedy 10 a více, cyklus se vůbec neprovede, v případě, že je menší, cyklus se provede tolikrát, dokud bude podmínka splněna a na samostatnou řádku vypíše číslo, které proměnná obsahuje.

```

24 -   while (x < 10) {
25     x++;
26     System.out.println(x);
27   }

```

Obrázek 7: Cyklus while. Zdroj: Vlastní zpracování.

Příkaz `do-while` se od předchozího jmenovaného (`while`) liší tím, že proběhne alespoň jednou. Podmínka se tedy testuje až po průchodu cyklem, nikoliv na začátku. Je to stejně jako příkaz `while` vhodný příkaz, pokud na počátku nevíme, kolikrát má příkaz proběhnout [22].

Př.: Podobně jako u předchozích příkladů, nad cyklem je definice třídy, metody `main` a proměnné `i`, která je typu `int`. Příkaz v závorkách po syntaxi `do` se provede a následně se kontroluje, zda je splněna podmínka. Pokud ano, příkaz se provede znovu, pokud ne, pokračuje se v kódu dál. Opět se na jednotlivé řádky vypíše hodnota proměnné `i`, tolikrát, dokud je splněna podmínka.

```

29 -   do {
30     i++;
31     System.out.println(i);
32   } while (i < 10);

```

Obrázek 8: Cyklus do-while. Zdroj: Vlastní zpracování.

Java je objektovým programovacím jazykem. V objektovém jazyce je základním stavebním kamenem Třída (`class`). Třída představuje soubor proměnných (nebo lépe dat – mohou tam být i konstanty) a podprogramů. Proměnným se říká členské proměnné (member variables) nebo též datové složky a atributy a je v nich uložen stav objektu. Podprogramům se říká metody (methods) a manipulují s členskými proměnnými, čímž mění stav objektu. Metody tedy popisují schopnost (dovednosti, vlastnosti,...) objektu [22].

Třída je ale jen jakási šablona (objektový typ) a sama o sobě nemá přidělenou žádnou paměť, tedy nedá se pracovat ani s proměnnými, ani s metodami (pokud před nimi nebylo klíčové slovo `static`) [22].

Objekt (object) je datový prvek, který je vytvořen podle vzoru třídy. Často se říká, že objekt je instance třídy (instance) a často se termíny objekt a instance vzájemně zaměňují [22].

Podle jednoho vzoru třídy lze vytvořit libovolné množství objektů (instancí), které mají „dovednosti“ – tj. metody, a mohou mít navzájem různé nebo i stejné stavy, tj. obsahy členských proměnných [22].

1.8 Databáze a SQL

Pro potřeby bakalářské práce postačí základní znalost databází, je dobré tedy začít definicí, která čtenáři stručně popíše, co to vlastně databáze je.

Databáze je definována jako kolekce logicky propojených dat, která jsou společně uložena [23].

Databáze je kolekce organizovaných informací, takže jsou jednoduše přístupné, mohou být jednoduše spravovány a aktualizovány [24].

Pokud víme, co je to databáze, můžeme přistoupit k otázce, co je to SQL, tedy Strukturovaný dotazovací jazyk.

Strukturovaný dotazovací jazyk SQL, jinak pojmenovaný též jako Strukturovaný anglický dotazovací jazyk SEQUEL, je relační dotazovací jazyk. Je to standardizovaný soubor příkazů, které mohou být použity ke komunikaci a správě relační databáze [23].

Strukturovaný dotazovací jazyk SQL je standardizovaný jazyk vyvinutý pro tvorbu interaktivních dotazů určených k aktualizaci databáze, jako je IBM DB2, nebo Microsoft SQL Server a databázové produkty od společností Oracle, Sybase a Computer Associates [24].

Pod pojmem databáze si tedy můžeme představit soubor tabulek s daty, která jsou spolu propojená, mají jasnou strukturu, lze je aktualizovat a lze z nich získávat informace pomocí dotazovacích jazyků jako je například SQL. V následujících řádcích se čtenář práce může dozvědět, jaké jsou základní SQL dotazy – příkazy, které lze nad databází provádět. Je velmi důležité, aby byly příkazy napsány syntakticky správně. I nepatrné chyby (například opomenutí definování typu hodnoty) mohou vést k neprovedení příkazu a výpisu chybové hlášky.

V následujícím příkazu může čtenář vidět syntaxi pro vytvoření tabulky `CREATE TABLE`, která bude mít název „navez_tabulky“ a bude obsahovat sloupce „Sloupec1“, „Sloupec2“, „Sloupec3“, „Sloupec4“, které budou mít definované typy (místo „typ_hodnoty“ uživatel vloží příslušný typ). V příkazu lze využít i syntaxi `WHERE`, která značí podmínku.

```
3 CREATE TABLE navez_tabulky (Sloupec1 typ_hodnoty, Sloupec2 typ_hodnoty,  
4 Sloupec3 typ_hodnoty, Sloupec4 typ_hodnoty);
```

Obrázek 9: Příkaz k vytvoření tabulky. Zdroj: Vlastní zpracování.

Tabulky jsou základním stavebním kamenem databáze. Každá tabulka se skládá z řádků a sloupců (atributů) dat v databázi. Příkaz `CREATE TABLE` definuje názvy tabulek a sloupců a také specifikuje typy dat, které se v jednotlivých sloupcích mohou nacházet [23].

Aby šlo tabulku nějak smazat, existuje příkaz `DROP`, kterým lze odstranit tabulku, případně pohled. Následující syntaxe říká, že v databázi přestane existovat tabulka s názvem „navez_tabulky“.

```
2 DROP TABLE navez_tabulky;
```

Obrázek 10: Vymazání tabulky. Zdroj: Vlastní zpracování.

Dalším příkazem je příkaz `SELECT`, který slouží k vybrání dat z tabulky nebo více tabulek. Z následující syntaxe vyplývá, že uživatel chce vybrat všechna data z tabulky pojmenované „navez_tabulky“, kde se nachází ve sloupci pojmenovaném „Sloupec2“ hodnota BP.

```
5 SELECT * FROM navez_tabulky WHERE Sloupec2 = 'BP';
```

Obrázek 11: Příkaz SELECT. Zdroj: Vlastní zpracování.

Příkaz `SELECT` je jediným příkazem, který lze zařadit do DQL (Data Query Language), tedy datový dotazovací jazyk. Příkaz `SELECT` koresponduje s projekční operací v relační algebře. Je používán k zobrazení atributů (sloupců) v žádaném formátu definovaném dotazem. V příkazu `SELECT` lze mimo jiné definovat seřazení řádků (příkaz `ORDER BY`), případně vytvořit skupiny ze záznamů pomocí příkazu `GROUP BY` [23].

Posledním z předvedených příkazů je příkaz `VIEW` - pohled, který reprezentuje virtuální tabulku, vytvořenou z dat uložených v databázi. Konkrétní příklad užitý v této práci představuje pohled z tabulky „navez_tabulky“, který je pojmenovaný jako „Pohled1“, znázorňuje „Sloupec1“ pojmenovaný jako „prvni_sloupec“ a „Sloupec3“ pojmenovaný jako „druhy_sloupec“.

```
13 CREATE VIEW Pohled1 AS SELECT Sloupec1 as prvni_sloupec, Sloupec3 as druhy_sloupec
14 FROM navez_tabulky WHERE Sloupec4 = '5';
```

Obrázek 12: Příkaz k vytvoření virtuální tabulky. Zdroj: Vlastní zpracování.

Pohled (`VIEW`) je pojmenovaná tabulka, reprezentovaná definicí jinak pojmenovanými tabulkami. Jde o virtuální tabulku, která je konstruovaná automaticky, a data jsou stále reálně uchována v původních tabulkách, ze kterých pohled vznikl. Operace `CREATE VIEW` definuje logickou tabulku, která vzniká z jedné nebo více existujících tabulek nebo pohledů [23].

2 Metodika práce

2.1 Představení společnosti AIMTEC a.s.

Tato bakalářská práce byla vypracována ve spolupráci s Plzeňskou společností AIMTEC a.s., která se zabývá vývojem, implementací a poradenstvím v oblasti informačních systémů se zaměřením hlavně na automobilový průmysl.

Společnost AIMTEC spolupracuje se světovými lídry v oboru vývoje informačních systémů, jako je například německá společnost SAP AG, pro kterou společnost AIMTEC působí jako dodavatel jejích řešení pro podniky v České republice. Dále společnost AIMTEC nabízí společnosti SAP AG služby svých konzultantů v oblasti EDI komunikace.

Společnost AIMTEC a.s. lze formálně rozčlenit na čtyři hlavní divize:

SAP – divize se zabývá implementací již zmiňovaného informačního systému (jeho modulů) společnosti SAP.

EDI – divize operuje v oblasti elektronické výměny dat a poskytuje podporu zákazníkům (v současnosti z divize vzniká nová divize OPERATIONS, která se bude specializovat na první úroveň podpory zákazníků v režimu 24/7). Divize EDI dále nabízí služby svých konzultantů jako externistů v jiných společnostech (společnost SAP AG, společnost TechData corp.,...). Divize EDI dále nabízí EDI jako službu prostřednictvím cloudového řešení, produktu ClouEDI.

ASPROVA – divize se zabývá implementací japonského informačního systému Asprova, který slouží hlavně k plánování výroby.

DCI – divize zabývající se vývojem a implementací in-house informačního systému společnosti AIMTEC s názvem DCIx. Informační systém obsahuje více modulů, jedním z nejvíce poskytovaných modulů je systém řízení skladu.

V současnosti společnost AIMTEC přesahuje roční obrat 200 milionů Kč (rok 2015), společnost stabilně roste a v posledních letech zvyšuje každoročně počet svých zaměstnanců o desítky procent. Většina zákazníků společnosti jsou zahraniční klienti, v České republice se společnost AIMTEC řadí mezi nejsilnější společnosti v oblasti informačních systémů na

trhu. Nejvíce zákazníků má společnost AIMTEC v Evropě, nicméně má zákazníky také například v Číně nebo v USA.

2.2 Vznik tématu práce

Téma práce vzniklo ze seznamu nápadů autora bakalářské práce, a byla dále probírána na pozdější konzultaci se zaměstnanci firmy AIMTEC, s panem Ing. Lukášem Rampou a panem Ing. Vlastimilem Šilhánem, kteří ve společnosti působí jako manažeři v oddělení elektronické výměny dat – EDI. Následující odstavec práce obsahuje seznam nápadů-témat práce, nad kterými bylo během konzultace uvažováno:

- Metoda TOC – Theory of Constraints – teorie omezení. Zjištění prostojů ve výrobním procesu, kolik osob přijde s určitým materiálem během výroby do styku, jak dlouho se s ním nic neděje. Data by šlo získat z informačních systémů – id skladníků, dopravců, a podobně. Vždy, když se s materiálem manipuluje, je oscanován jeho QR/čárový kód.
 - Téma bylo na konzultaci jejími účastníky zamítnuto z důvodu obtížnosti získání dat, která by mohla být analyzována. Dále kvůli rozmanitosti firem, které využívají informační systémy společnosti AIMTEC, a hlavně kvůli nekonzistentnímu přístupu na straně uživatelů informačního systému.
- Monitoring exportu jednotlivých druhů výrobků nebo materiálů podle lokalit (napříč více firmami). Data přínosná pro nově začínající podniky, které se chtějí specializovat na stejný typ produktu.
 - Nedostatek firem v portfoliu, nevhodné vzhledem k zaměření firmy AIMTEC. Zaměřením společnosti je hlavně automobilový průmysl, který je specifický tím, že firmy v odvětví o sobě vědí a jde většinou o velké firmy.
- Monitoring dodacích lhůt a cen konkrétních dodavatelů – poradenství pro firmy, které s těmito dodavateli spolupracují, nabídka rychlejší/levnější konkurence.
 - Podobné řešení již vymyslela společnost ODETTE International, která operuje v oblasti informačních systémů pro logistiku a výrobu. Na B2C úrovni funguje například jako internetové srovnávače typu heureka.cz.

- Monitoring produktů, které uživatelé systémů společnosti AIMTEC vyrábí. V systému by mohlo jít vyčíst, o jaké součástky se jedná, kolik jich kdo vyrábí a za jakou cenu. Lze tak jednoduše sledovat, o jaké výrobky je zájem, kde je pro ně uplatnění.
 - Zde jde o velmi kontroverzní řešení, společnosti, které jsou zákazníky společnosti AIMTEC by zřejmě neposkytly souhlas s užitím těchto analýz v praxi.
- Sledování odvolávek napříč různými dodavateli jedné automobilky. Jak moc je automobilka přesná v predikcích výroby směrem k jejím dodavatelům, kolik materiálu do svojí výroby skutečně nakoupí, a kolik původně objedná.
 - K tomuto tématu existuje větší množství dat, která lze k práci využít. Společnost AIMTEC je schopna poskytnout anonymizovaná data několika firem, které jsou dodavateli větší automobilky působící v České republice. Výstupní data (grafy) by bylo možné nabídnout dodavatelům automobilky, a poskytnout jim tak informaci o tom, jak je jejich odběratel přesný v objednávání materiálu. Následně by podle získaných informací mohli optimalizovat svoji výrobu.
- Monitoring času vystavování dokladů. Zjišťování časů, kdy je větší aktivita (je odesíláno hodně faktur/jiných dokladů), a kdy naopak není aktivita skoro žádná. Možností je rozšíření o nákupy po novém roce – dálniční známky, licence na SW atd.
 - Vzhledem k automatizaci řešení není důvod. Doklady jsou vytvářeny tehdy, když je potřeba. Dálniční známky a nákup licencí nejsou tak časově náročné aktivity, aby se je vyplatilo optimalizovat dle času.
- Investice do reklamy v daném oboru – kdy, jaký druh, za kolik? Téma je založeno na porovnávání investic napříč odvětvím. Mohla by zde být možnost integrace reklamy reklamních agentur s největší efektivitou, nejnižší cenou a podobně.
 - Záměna příčin a následků. Každá firma je v jiné situaci, proto nelze srovnávat investice do reklamy. Takováto služba by navíc spíše obtěžovala nevyžádanou reklamou na reklamní agentury.
- Sledování setrvačnosti krize – je možné, že šlo předpovědět, že bude krize podle chování firem a společnosti? Existovaly v období před krizí nějaké ukazatele z výroby, které nasvědčovaly tomu, že se blíží krize? Jak se promítají hospodářské výsledky firem a prognózy výroby do grafů HDP ekonomik?

- Zajímavé téma s těžko získatelnými daty a velmi nejistým výsledkem. Dat by muselo být velmi mnoho, abychom mohli s rozumnou přesností říci, zda lze výstup považovat za legitimní. Toto téma počítá s větším rozsahem než je rozsah jedné bakalářské práce.

2.3 Výběr tématu práce

Vzhledem k dostupnosti dat ke sledování, know-how firmy AIMTEC a celkové uchopitelnosti tématu se autor společně se zaměstnanci společnosti AIMTEC rozhodli, že nevhodnějším tématem pro bakalářskou práci bude téma vycházející ze sledování dat v odvolávkách. Téma by mělo být využitelné v praxi, mělo by být splnitelné v rámci vyhrazeného času pro bakalářskou práci, a zároveň by mělo být dostatečně zajímavé jak pro studijní, tak pro praktické účely.

Konkrétním výstupem práce by měl být parser, který je využit k extrakci dat ze získaných souborů obsahujících zprávy, návrh vhodného uložení získaných dat a následné vytvoření grafu, který by mohl být integrován v některém z produktů společnosti AIMTEC.

2.4 Zapojení ERP systémů dodavatelů a odběratele

V této bakalářské práci se soustředíme především na komunikaci mezi ERP systémem výrobního podniku, který v práci vystupuje jako odběratel, a mezi ERP systémy jeho dodavatelů, kteří jsou taktéž výrobními podniky a dodávají materiál pro odběratelovu výrobu. Vybereme-li konkrétní příklad modulů, které jsou do komunikace zapojeny, jde jistě o modul plánování výroby na straně dodavatele i na straně odběratele, nebo modul řízení skladu dodavatele, jelikož se budeme zabývat takzvaným odvoláváním, tedy predikcemi odběru materiálu ze strany odběratele. Samozřejmě je do procesu zapojeno více modulů obou ERP systémů, nicméně tyto dva jsou v rámci tématu zapojeny přímo.

Odběratel podle množství zakázek (tudíž požadavků na výrobu) stanovuje dopředu v daném časovém horizontu množství výrobků, které musí vyrobit, aby uspokojil poptávku. Je žádoucí, aby se tato informace propadla k dodavateli, aby se i on mohl připravit na to, kolik bude muset vyrobit výrobků, a kolik na to bude potřebovat materiálu od svých dodavatelů. Vzhledem k tomu, že náš odběratel je velká nadnárodní společnost zabývající se výrobou

automobilů, jde o společnost se silnou pozicí na trhu a velkým kapitálem. Velká firma jako je automobilka s několikamiliardovým obratem si tedy může diktovat pravidla a stanovovat pro své dodavatele dlouhodobé kontrakty, které musí dodavatelé plnit. V případě, že je dodavatel nesplní, plynou ze vzniklých problémů vysoké finanční sankce, které musí dodavatel odběrateli zaplatit.

2.5 Použitý software

V práci je potřeba číst zprávy, které jsou ve formátu VDA, k čemuž je možné využít různé druhy softwaru. Díky tomu, že autor má přístup k upravené verzi editoru Vim, který si modifikovala sama společnost AIMTEC a.s. právě ke čtení různých zpráv v EDI formátech, na tuto problematiku bude využito právě tohoto softwaru.

Vzhledem k tomu, že zprávy byly v jednotlivých složkách, bylo potřeba tyto zprávy ručně vykopírovat a k tomu byl využit příkaz MS DOS přes příkazovou řádku operačního systému Microsoft Windows, na kterém také všechny programy běží.

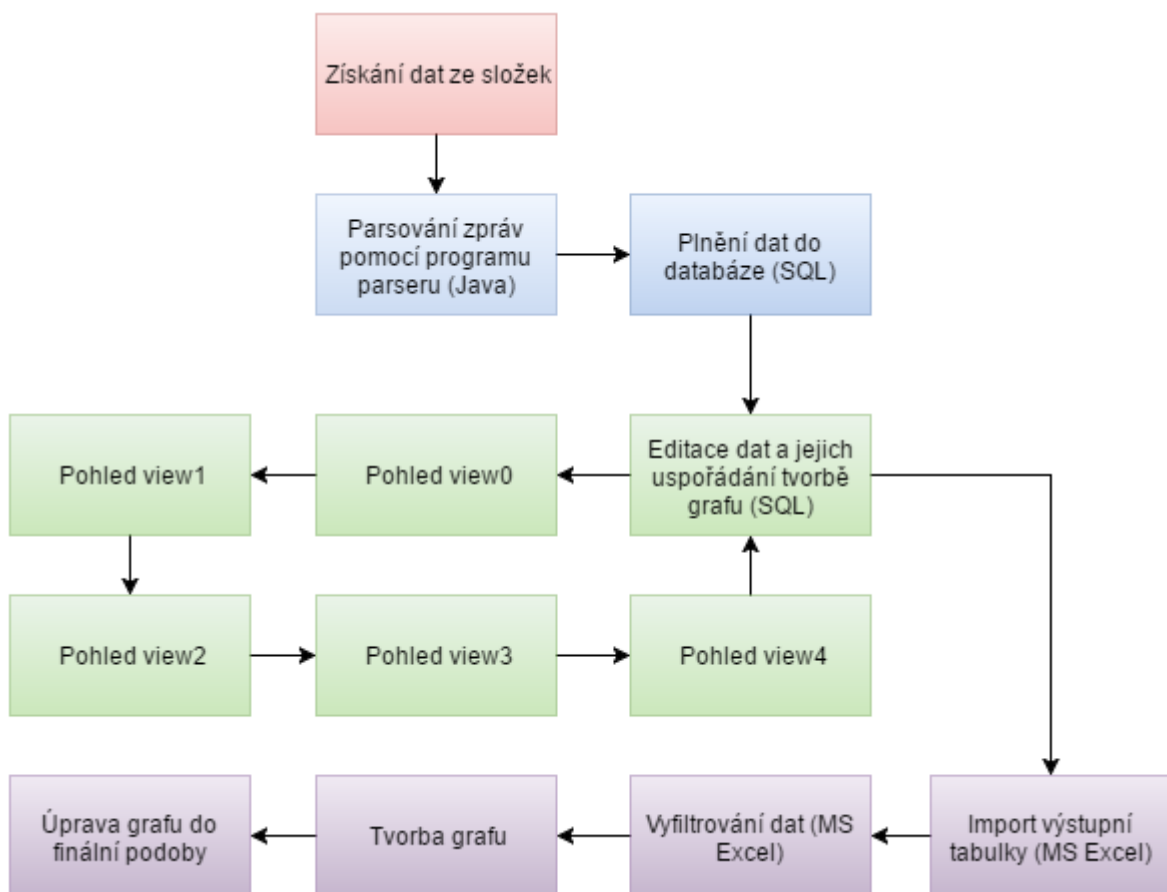
Od počátku bylo jasné, že zprávy bude nutné nějak procházet, k čemuž autor práce zvolil silný a velmi robustní jazyk Java od společnosti Oracle, společně s balíkem knihoven, které jsou volně dostupné k programování aplikací v tomto jazyce.

K parsování dat byl využit program v jazyku Java, který byl vyvinut ve vývojovém prostředí Eclipse. Ve vývojovém prostředí je také možné kód debugovat, tedy postupně procházet za účelem nalezení chyb. K následnému použití databáze bylo nutné databázi propojit s tímto vývojovým prostředím, k čemuž bylo využito JDBC ovladače.

Z počátku byl ke všem operacím nad surovými daty využit Microsoft Excel, který je součástí balíku Microsoft Office. Jeho úkol byl postupně nahrazen databázovým serverem Microsoft SQL Server 2014. V databázi byla provedena řada operací nad tisíci záznamů získaných ze zpráv pomocí parseru. Již zmiňovaný Microsoft Excel stále zůstává využit při závěrečné tvorbě grafu, který je tvořen z tabulky importované z databázového serveru.

2.6 Posloupnost procesů

Od vstupních zpráv až po konečný sestrojený graf je třeba sestavit procesy tak, jak jdou postupně za sebou. Zjednodušený postup znázorňuje diagram na následujícím obrázku. Barevně je zde rozlišeno, kde daná akce probíhá. Červená barva je označením pro příkazovou řádku, modrá označuje parser, zelená procesy probíhající v databázi a fialová označuje procesy v MS Excel.



Obrázek 13: Diagram procesů. Zdroj: Vlastní zpracování.

Pro srozumitelnost informací v předchozím diagramu je v následujících odstavcích shrnuta struktura procesů slovně.

Na počátku je třeba získat data z jednotlivých složek. Je nutno říci, že data jsou anonymizovaná, takže nedojde k narušení interních informací společností zapojených do zkoumání. Data jsou pomocí MS DOS příkazů psaných do příkazové řádky operačního

systemu Microsoft Windows vykopírována do jednoho adresáře a pojmenována ve správném formátu tak, aby je bylo možné číst parserem.

Když jsou data připravena v adresáři, odkud jsou parsována, může být spuštěn parser v jazyku Java (v rámci bakalářské práce pouze z vývojového prostředí). V průběhu parsování zpráv jsou data plněna do databáze.

Pro editování databáze je potřeba spustit Microsoft SQL Server, kde je otevřeno všech pět vytvořených pohledů, a jsou postupně od 0 až do 4 provedeny. Správné pořadí je důležité pro správné provedení operací nad aktuálními údaji.

Když jsou provedeny všechny pohledy, je možné otevřít kancelářský nástroj Microsoft Excel, kde je třeba naimportovat tabulku vytvořenou pohledem `view4`. Tato data jsou poté uživatelem odfiltrována na zvolený týden difference, jsou vybrána všechna vyfiltrovaná data k tvorbě grafu a je vytvořen graf.

2.7 Sledování odvolávek napříč dodavateli

V práci je důležité zmínit, proč bylo zvoleno sledování odvolávek odesílaných různým dodavatelům jedním a tím samým odběratelem. K tomuto rozhodnutí vedla předběžná analýza, na prvním vzorku zpráv, která byla tvořena ručně pomocí kontingenčních tabulek v excelu. Autor společně se zaměstnanci společnosti AIMTEC dospěli k názoru, že automobilka odesílá všem dodavatelům podobně nepřesné předpovědi, tudíž je vhodné přesnost sledovat plošně. Dalším důležitým faktem je, že pokud bude výstupní graf využit v praxi, měl by posloužit i novým dodavatelům automobilky.

3 Vlastní řešení

3.1 Vstupní data

Společnost AIMTEC poskytla autorovi práce vzorek anonymizovaných dat (47 zpráv VDA4905). Ukázková zpráva je přiložena v bakalářské práci v seznamu příloh, jako příloha B.

Ke zpracování tohoto prvního balíku dat se nabízely dvě možnosti. První možností bylo vypsát všechna data a množství ručně do excelové tabulky, vhodné je uvést také datum, kdy byla zpráva vystavena a další údaje sloužící k identifikaci dodavatele i odběratele. Takto by ale práce s daty trvala možná desítky hodin, tabulka by měla odhadem několik tisíc řádků a je téměř jisté, že by ve výstupním excelovém souboru byly překlepy a chyby plynoucí z nepozornosti. Druhou možností bylo naprogramovat parser, který bude data z jednotlivých zpráv automaticky extrahovat. Autor práce se rozhodl pro druhou možnost, už jen proto, že počet vstupních zpráv není finální. Dalším důvodem byl fakt, že pokud budou procesy automatizovány, bude možné parser použít jako základ pro tvorbu grafů, které lze využít k rozvoji produktů společnosti AIMTEC.

Po zpracování prvního balíku dat pomocí parseru (viz kapitola 3.2) autor obdržel další balík dat, která jsou vztažena k dalším dvěma odběratelům automobilky. Celkem se tedy jednalo o data třech zákazníků z období mezi koncem roku 2013 a srpnem roku 2015. V součtu měl autor k dispozici celkem 3077 zpráv v samostatných složkách. Jednotlivé soubory zpráv si autor práce pomocí příkazové řádky vykopíroval do jedné složky příkazem na následujícím obrázku.

```
for /R c:\zdroj\ %f in (*.*) do copy "%f" c:\cíl\
```

Obrázek 14: Příkaz na extraktci souborů ze složek (příkazová řádka). Zdroj: Vlastní zpracování.

V balíku souborů se také vyskytovaly duplicitní soubory, které by zásadně zkreslovaly výstupní grafy, kdyby nebyly odfiltrovány. Odfiltrování duplicitních souborů je řešeno v parseru.

3.2 Parser

K extrakci dat ze samotných zpráv se autor práce rozhodl využít programovacího jazyka Java, a ve vývojovém prostředí Eclipse postupně vytvořil parser, který zvládne ze vstupního adresáře načítat zprávy v unifikovaném formátu a postupně je procházet podle jasně určených pravidel.

3.2.1 Ošetření výskytu duplicitních zpráv

Jak již bylo zmíněno, je třeba ošetřit, aby nedošlo ke zpracování duplicitních souborů zpráv. Řešením je poměrně jednoduchý kód, který zajistí, aby se při načtení každé zprávy zkontrolovaly klíčové identifikační údaje uvnitř hlavičky zprávy (Segment „511“), konkrétně datum přenosu zprávy, číslo dodavatele, a přenosové číslo transakce (unikátní, ale pouze pro jednoho dodavatele). Dále je pak využito prvního segmentu „512“, který obsahuje identifikační číslo materiálu. Segmentu „512“ se autor rozhodl využít proto, že se tím zvýší pravděpodobnost, že zpráva bude unikátní i při použití více dat od různých dodavatelů. Řetězec, který vznikne sloučením všech těchto hodnot v tomto pořadí, se porovnává s řetězcí všech předchozích zpráv, které jsou uloženy do pole řetězců. Pokud je nalezena duplicita, zpráva je z dalšího zpracování vyřazena.

3.2.2 Postup čtení zprávy

Zprávy jsou pojmenovány v unifikovaně zvoleném formátu „message(X).vda“, kde „X“ je číslo zprávy udělené autorem práce. Takovéto pojmenování zpráv je v operačním systému Windows poměrně triviální, stačí označit všechny zprávy, a přejmenovat je kliknutím na první zprávu, kterou pojmenujeme „message(1).vda“. Průzkumník Windows pak zbytek práce provede za uživatele.

Zpráva se skládá ze segmentů označených trojčíslím „511“ až „519“. Dále je ve zprávě na jedné řádce přesně 128 znaků, kde jsou nevyužité pozice vyplněny mezerami nebo nulami, takže je jisté, že ve zprávě budou všechny řádky stejně dlouhé, a bude tedy dodržen neměnný formát. Zpráva může obsahovat náhodný počet řádek, musí být však zachována struktura definovaná standardem. Díky jasně dané struktuře se dá zprávou jednoduše procházet. Nejdříve je načten první řádek (hlavička). Po načtení první řádky začne cyklus while, který

je prováděn nad jednou zprávou tak dlouho, dokud není dosaženo konce zprávy (tedy dokud není načten řádek patičky).

V tabulce číslo 11 je ukázka údajů, které jsou čteny z jednotlivých řádků. Ve zprávě je více informací, které by šly extrahovat, nicméně vzhledem k nepotřebnosti v této bakalářské práci zpracovány nebyly.

Segment	Extrahovaná data
„511“	Datum vystavení zprávy, ID dodavatele, přenosové číslo transakce
„512“	ID Materiálu, číslo zprávy (počítadlo), ID závodu, místo vykládky, číslo objednávky, typ materiálu („S“ / „E“)
„513“	Datum požadovaného dodání (může být ve formátu dní/týdnů, nebo jde o kód), množství, oboje 5x
„514“	Datum požadovaného dodání (může být ve formátu dní/týdnů, nebo jde o kód), množství, oboje 8x

Tabulka 11: Tabulka s daty, která jsou extrahována parserem. Zdroj: Vlastní zpracování.

Při čtení segmentů s označením „513“ a „514“ se data rovnou odesílají do metody sloužící k výpisu, společně se všemi údaji, které je třeba vypsát.

3.2.3 Výpis extrahovaných dat

O výpis, se podobně jako o čtení, stará samostatná metoda spolu s dalšími metodami pro specifické případy.

V metodě je několik podmínek sloužících ke čtení šesticiferných kódů (tabulka číslo 9). Pokud se datum požadovaného dodání rovná kódu 444444, je tento datum nastaven na hodnotu rovnou datu vystavení zprávy, protože jde o kód definující, že materiál je požadovaný ihned. Pro kódy 000000, 222222 a 333333 je hodnota nevyužita nebo přepsána na prázdnou hodnotu, protože tyto požadavky nejsou předmětem zkoumání. Podmínka pro kód 999999 není implementována, protože ve zprávách není využíván, nicméně je to jedna z možností rozšíření programu. Pro kód 555555 je v parseru speciální globální podmínka, která je nastavena jako pravdivá v případě, že se tento kód v řádku objeví. Všechny hodnoty za tímto kódem jsou od jeho výskytu brány jako týdenní předpovědi.

Po ošetření výjimek z předchozího odstavce jsou v metodě obsaženy dvě podmínky. Jedna pro denní předpovědi, druhá pro týdenní předpovědi. Tyto podmínky nastaví data pomocí pomocných metod do požadovaného formátu, který lze vložit do databáze tak, aby byl pro všechny případy v rámci jednoho typu data stejný. Výstupem z těchto metod je zavolání metody `printTo()`, do které již vstupují data v požadovaném formátu. Konkrétně se jedná o číslo zprávy (počítadlo) dodavatele, datum vystavení zprávy ve formátu `DD.MM.YYYY`, datum zprávy ve formátu `YYMMDD`, datum předpovědi v původním formátu ve zprávě (buď týdenní, nebo denní), datum předpovědi ve formátu `MM/YYYY`, požadované množství, ID dodavatele, ID materiálu, ID závodu, ID místa vykládky, číslo objednávky, kód říkající, zda jde o sériový produkt či prototyp, datum předpovědi ve formátu `DD.MM.YYYY` pro denní a ve formátu `WW/YYYY` pro týdenní předpovědi.

V metodě `printTo()` proběhne poslední úprava dat, jedná se již pouze o formát data. Na toto je využita metoda `setFormatToDate()`. V programu se počítá s budoucím využitím na grafy s nejmenší časovou jednotkou jedno čtyřtýdenní, proto je v metodě `printTo()` počítáno i s daty ve formátu měsíců (zjednodušeno na 4 týdny), k čemuž je opět využita převáděcí metoda `setFormatToDate()`. Je důležité zmínit, že v databázi jsou všechna data využívána k tvorbě grafu ve formátu týdnů, nebo čtyřtýdenní, což znamená, že datum vždy odkazuje na pondělí – první den daného týdne. Po přidání dalších parametrů, se všechna data pro jeden řádek vloží pomocí třídy `PreparedStatement` a ovladače `JDBC` do databáze, která je vytvořena při startu programu. V databázi je po startu programu vytvořena databázová tabulka s předdefinovanými sloupci, které se pomocí metody `printTo()` naplní daty. Pro testovací účely je v metodě `printTo()` nastaveno vypisování údajů do konzole a do souboru typu `.csv`. Ve výsledné databázi v databázovém serveru `Microsoft SQL Server 2014` existují v oddělených sloupcích hodnoty, které popisuje tabulka číslo 12.

3.2.4 Propojení parseru s databází

Pro propojení databáze (SQL) s parserem (Java) je využito ovladače `JDBC` verze 4.0.4621.201, a třídy `PreparedStatement`, které jsou volně ke stažení na webu společnosti `Microsoft` [25].

Propojení vývojového prostředí Eclipse s ovladačem JDBC je poměrně jednoduché, stačí v nastavení vývojového prostředí Eclipse nainportovat ovladač do Java Build Path a následně ho pomocí několika řádků kódu využít ve vlastním programu. Pro zapisování do databáze je nutné, aby byla připojena. K navázání spojení je třeba zadat URL adresu databáze spolu s přihlašovacím jménem a heslem. Pro připojení je v parseru využito třídy `DriverManager`.

Zapisování do databáze je prováděno pomocí třídy `PreparedStatement`, Opět se jedná o jednoduchý postup, pomocí kterého uživatel předepíše SQL dotaz, který chce provést a následně ho pomocí metody `executeUpdate()` provede.

3.2.5 Možnosti vylepšení parseru

Program parseru určitě není dokonalý, a je zde jasně patrných hned několik možností, jak jej lze zrychlit a zefektivnit:

- Program je tvořen pouze jednou třídou, což omezuje jeho rozšiřitelnost a použitelnost v dalších aplikacích. Převedení kódu do lépe strukturované podoby by určitě pomohlo jeho čitelnosti.
- Pokud jde o rychlost parsování zpráv, největším problémem je postupné zapisování do databáze. Tento problém lze vyřešit vytvořením komplexního výstupu již uvnitř parseru a následným jednorázovým vložením dat do databáze.
- Možnou nevýhodou může být častá změna formátu, hlavně u proměnných obsahujících datum. Aktuální nepřehlednost je daná hlavně tím, že program vznikl postupně. Dále také proto, že veškeré informace o tom, která data bude potřeba extrahovat, nebyly dostupné, a z důvodu prvotní nezkušenosti autora nebyly brány v potaz ihned od začátku procesu. Vznikalo tak několik verzí, například od začátku prací vůbec nebylo jisté, zda bude využito databáze, jejíž následné využití mnoho postupů ulehčilo. Mnoho částí kódu však bylo třeba změnit tak, aby šlo data do databáze vkládat. Při případném přepsání programu lze jistě vymyslet jednodušší a efektivnější převody.
- Implementace měsíčních dat – v současnosti je pro tuto variantu využíváno takzvané „čtyřtýdenní“, hlavně z toho důvodu, aby bylo možné bez odchylky porovnávat,

stejně období a stejná data s rozdílnou jemností. V případě, že by byly nyní implementovány reálné měsíce, došlo by k znehodnocení výstupních grafů, protože aktuální nejmenší časový úsek je jeden týden, tudíž by data byla špatně rozdělena. Řešením je stanovit jako nejmenší časovou jednotku jeden den, a v případě předpovědí na rozpětí týdnů nedělit předpovídané množství počtem týdnů, ale počtem dní. Takto by vzniklo několikanásobně větší množství záznamů v databázi, ale data by byla přehlednější a lépe uchopitelná z pohledu dalšího zpracování.

3.3 Databázové operace a SQL dotazy

Protože s parsováním elektronických zpráv, kterých je velké množství, souvisí potřeba tato data rozumně uchovávat. Jako vhodné úložiště dat byla vybrána databáze.

Pokud zde chceme rozepsat postupné kroky, které vedly k rozhodnutí využít databázi, jako první by mělo být zmíněno, že v rámci prvního testování parseru na prvních 47 zprávách typu VDA4905 bylo použito výpisu na obrazovku (do konzole ve vývojovém prostředí Eclipse). Následně bylo pro přehlednost, a z důvodu jednodušší práce se získanými daty, rozhodnuto vkládat data zároveň do souboru CSV, který je možné otevřít v programech jako je MS Excel. Data lze jednoduše vkládat do určených buněk pomocí středníku, který značí, že je potřeba přeskočit do vedlejšího sloupce. Řádky jsou zde řešeny stejně jako u výpisu do konzole prostým výpisem s odřádkováním po zadání poslední hodnoty na řádce.

S rostoucím objemem dat používaných v práci po získání druhého balíku obsahujícího 3077 zpráv (včetně duplicit), bylo třeba zvolit elegantnější řešení pro jejich uchování, následnou restrukturalizaci a propojení. Příprava dat pro tvorbu jednoho grafu v této fázi trvala několik hodin, proto autor práce společně s podporou společnosti AIMTEC zvolil použití databáze, konkrétně Microsoft SQL Server 2014 v bezplatné verzi pro nekomerční využití. Po instalaci a založení účtu stačilo MS SQL Server propojit s programem parseru (Java), k čemuž byl využit JDBC ovladač.

3.3.1 Naplnění databáze

Před naplněním databáze daty bylo potřeba vytvořit tabulku se sloupci s vhodnými datovými typy. Díky Javovské třídě `PreparedStatement` se podařilo vložit do databáze tabulku,

kteřá obsahuje ve finální verzi celkem 19 sloupců, ve kterých jsou obsaženy všechny potřebné hodnoty. Celkový výčet sloupců s jejich příkladovými hodnotami a datovými typy je možné vidět v tabulce číslo 12.

Název sloupce	Typ (velikost)	Příklad (denní)	Příklad (týdenní)
Message_number	int	2	1093
Message_cust_ID	int	14	14
Supplier_ID	varchar(30)	1000872802	1000872802
Material_ID	varchar(50)	1303214329	1683791916
Discharge_place_ID	varchar(10)	1859062959	1859062959
Plant_ID	varchar(4)	8747	8747
Order_number	varchar(20)	439848904	439848904
Requirement_code	varchar(2)	S	S
Message_date	date	2013-11-14	2013-11-14
Message_week	date	2013-11-11	2013-11-11
Prediction_week	date	2013-12-30	2014-06-23
Quantity	numeric(28,0)	864	575
Message_month	date	2013-11-04	2013-11-04
Prediction_month	date	2013-12-30	2014-06-16
Difference_week	int	7	31
Difference_month	int	1	7
Date_type	varchar(5)	day	week
Prediction_date	date	2014-01-03	2014-06-23
Original_prediction_date	varchar(7)	140103	140026

Tabulka 12: Tabulka obsahuje příklad výstupních dat v jednotlivých sloupcích (zde řádcích). Zdroj: Vlastní zpracování.

Tvorba tabulky probíhá na začátku programu parseru, kdy je navázáno spojení s databází. Nejdříve je vymazána stará tabulka pomocí SQL příkazu `DROP TABLE`, pokud tabulka v databázi existuje. Následně je příkazem `CREATE TABLE` vytvořena nová tabulka se stejnými parametry a je připravena k naplnění.

Vlastní naplnění databáze se provádí pomocí SQL příkazu `INSERT`, který se provede pro každý řádek tabulky zvlášť. Podobně jako u příkazu k vytvoření tabulky je zde využito třídy `PreparedStatement` (Java), díky které je vytvořen „prázdný“ SQL příkaz `INSERT`. Tento příkaz je poté naplněn hodnotami nashromážděnými v metodě `printTo()` v parseru a následně se provede. Takto vše proběhne pro každou dvojici předpovídaného data dodání a předpovídaného dodaného množství (celkem 253053 krát), samozřejmě pouze pro

neduplicitní zprávy (celkem 2650 souborů) – předchozí údaje jsou vztaženy k druhému balíku zpráv (3077 souborů).

```
Driver name: Microsoft JDBC Driver 4.0 for SQL Server
Driver version: 4.0.4621.201
Product name: Microsoft SQL Server
Product version: 12.00.2269
Deleting existing database table
Creating new database table
There were 2650 messages with 253053 items inserted into BP_table.
Program was performing for 88161 ms, which is approx. 1 min 28 sec.
```

*Obrázek 15: Kontrolní výpis v konzoli po skončení parseru a naplnění dat do databáze.
Zdroj: Vlastní zpracování.*

3.3.2 Logika příkazů nad databází

Práce nad databází je rozdělena na pět logických bloků. Hlavním důvodem je logické rozdělení jednotlivých kroků, které je nad databází žádoucí provést, aby bylo ve výsledku dosaženo tabulky obsahující tři sloupce, ze které lze později jednoduše vytvořit graf.

Prvním sloupcem tabulky je datum předpovědi „Prediction_week“, tedy datum, na kdy je poptáváno požadované množství ve formátu data akceptovaného programem Microsoft Excel. Ve druhém sloupci je uveden poměr požadovaného množství v den předpovědi ku požadovanému množství v den poslední předpovědi před požadovaným termínem doručení materiálu (bráno jako skutečně dodané množství). Pokud tedy odběratel (který dostává odvolávky – VDA4905) nadhodnotí odhad požadovaného množství na daný termín, bude číslo ve druhém sloupci větší než 100. Ve třetím sloupci je zobrazena týdenní diference, jinými slovy počet týdnů, které uběhnou od odeslání zprávy s předpovědí potřeby a mezi odesláním poslední zprávy (bráno jako skutečně dodané množství). Je důležité zmínit, že data jsou vždy vztažena k celému týdni. To znamená, že udávaná množství z jednotlivých zpráv, která jsou odeslána během jednoho týdne a cílí na určitý datum dodání, jsou sečtena, a z nich je pak počítán poměr určující přesnost předpovědi.

Aby vznikla ze surových dat naplněných do databáze výše popsaná tabulka, je nad nimi potřeba udělat hned několik operací. V následujících odrážkách je vysvětleno pět logických bloků, z nich každý sestává z jednoho pohledu.

- V pohledu pojmenovaném jako „view0“, je nutné odfiltrovat duplicitní hodnoty. To, že se v databázi vyskytují duplicitní záznamy, je způsobeno aktualizacemi zpráv ze strany odběratele o další druh materiálu v rámci jednoho dne. Počítadlo zpráv není změněno, protože jde o identickou odvolávku na určitý materiál, ale fakticky se soubor zprávy změní, takže zpráva není v parseru zachycena kontrolou na duplicitní zprávy. Funkcionalita nultého pohledu je tedy taková, že jsou vybrány všechny záznamy v databázi, kde se rovnají hodnoty s jiným záznamem (syntaxe GROUP BY), a následně jsou všechny duplicitní řádky sloučeny v jeden řádek. Tímto jsou odfiltrovány všechny duplicity, a je možné s tabulkou pracovat dále. Struktura tabulky po této operaci zůstává takřka nezměněná, jediný rozdíl je ten, že v ní již není obsaženo číslo zprávy definované autorem. Toto však není pro budoucí operace překážkou, protože tento druh identifikace se již dále nevyužívá. Po vykonání operace nad druhým balíkem zpráv zůstane z celkem 253053 řádků pouhých 206107 řádků.

	Message_cust_ID	Supplier_ID	Material_ID	Discharge_place_ID	Plant_ID	Message_date	Message_week	Prediction_week	Quantity	Difference_week	
1	1	1000872802	1038270329	1859062959	8747	2014-10-11	2014-10-06	2014-10-27	1122	3	
2	1	1000872802	1038270329	1859062959	8747	2014-10-11	2014-10-06	2014-11-03	990	4	
3	1	1000872802	1038270329	1859062959	8747	2014-10-11	2014-10-06	2014-11-10	924	5	
4	1	1000872802	1038270329	1859062959	8747	2014-10-11	2014-10-06	2014-11-17	594	6	
5	1	1000872802	1038270329	1859062959	8747	2014-10-11	2014-10-06	2014-11-24	462	7	
6	1	1000872802	1038270329	1859062959	8747	2014-10-11	2014-10-06	2014-12-01	792	8	
7	1	1000872802	1038270329	1859062959	8747	2014-10-11	2014-10-06	2014-12-15	1584	10	

Query executed successfully. MICHAL-NTB\BP (12.0 RTM) MICHAL-NTB\Michal (54) BP 00:00:02 206107 rows

Obrázek 16: SELECT s výtahem sloupců, virtuálního pohledu číslo 0 obsahujícího anonymizovaná data. Zdroj: Vlastní zpracování.

- V následujícím pohledu „view1“ jsou vybrány všechny záznamy, které mají stejný požadovaný týden dodání „Prediction_week“ a stejnou hodnotu difference „Difference_week“. Z těchto odvolávek jsou vybrány pouze poslední verze, tedy poslední zprávy v jednom týdnu (pro měsíce/čtyřtýdenní je třeba změnit podmínku) pro daný cílový den dodání, (toto je zaručeno pomocí pořadového čísla zprávy určeného dodavatelem materiálu – „Message_cust_ID“), pro daný závod „Plant_ID“, kam je materiál odvoláván, pro daného dodavatele „Supplier_ID“ a pro dané ID materiálu „Material_ID“. Tento pohled pro druhý balík zpráv (celkem 206107 řádků v databázové tabulce) vytvoří druhou virtuální pohledovou tabulku se zredukovaným počtem řádků – odfiltrované zastaralé odvolávky v rámci jednoho týdne (celkem 177595 řádků – tedy

přibližně o 30000 řádků méně než obsahují surová data bez duplicitních vstupních zpráv).

	Supplier_ID	Material_ID	Discharge_place_ID	Plant_ID	Message_date	Message_week	Prediction_week	Quantity	Difference_week
1	676705994	750728865	31879700	6134	2015-08-13	2015-08-10	2016-04-25	40	37
2	676705994	750728865	31879700	6134	2015-08-06	2015-08-03	2016-04-18	40	37
3	676705994	750728865	31879700	6134	2015-08-13	2015-08-10	2016-04-18	40	36
4	676705994	750728865	31879700	6134	2015-07-30	2015-07-27	2016-04-11	40	37
5	676705994	750728865	31879700	6134	2015-08-06	2015-08-03	2016-04-11	40	36
6	676705994	750728865	31879700	6134	2015-08-13	2015-08-10	2016-04-11	40	35
7	676705994	750728865	31879700	6134	2015-07-23	2015-07-20	2016-04-04	40	37

Query executed successfully. | MICHAL-NTB\BP (12.0 RTM) | MICHAL-NTB\Michal (57) | BP | 00:00:02 | 177595 rows

Obrázek 17: SELECT virtuálního pohledu číslo 1 obsahujícího anonymizovaná data. Zdroj: Vlastní zpracování.

- Ve druhém pohledu jsou sečtena všechna objednaná množství pro každou kombinaci týdenní diference „Difference_week“, týdne předpovědi „Prediction_week“, materiálu „Material_ID“, dodavatele „Supplier_ID“ a závodu „Plant_ID“. Sečtením všech hodnot podle této logiky je dosaženo bližšího propojení dat, která spolu souvisí (jsou to odvolávky určené na jeden cílový týden, ale zároveň jde o data fixovaná na jednotlivé dny v týdnu, tudíž je nelze přemazat a brát jen poslední údaj z týdne) a připravíme je k dalšímu párování. Tímto příkazem vznikne virtuální tabulka s celkovým počtem 116563 řádků. Ve výsledné tabulce jsou sloupce obsahující data o ID dodavatele „Supplier_ID“, ID materiálu „Material_ID“, týdenní diferenci „Difference_week“, týden předpovědi „Prediction_week“, ID závodu „Plant_ID“, sečtené množství „Quantity“ a informativní počítadlo počtu sečtených záznamů „Record_count“.

	Supplier_ID	Material_ID	Difference_week	Prediction_week	Plant_ID	Quantity	Record_count
1	36323457	747713257	17	2015-04-20	8747	1050	1
2	36323457	1420333436	23	2015-09-07	3031	193	1
3	1000872802	559478448	15	2014-06-09	6134	1840	1
4	1000872802	240592885	5	2015-02-16	8747	598	1
5	1000872802	1769581263	10	2015-03-23	8747	700	1
6	1000872802	1731268216	3	2014-06-16	8747	1200	1
7	1000872802	1573093244	24	2014-10-13	3031	320	1

Query executed successfully. | MICHAL-NTB\BP (12.0 RTM) | MICHAL-NTB\Michal (56) | master | 00:00:02 | 116563 rows

Obrázek 18: SELECT virtuálního pohledu číslo 2 obsahujícího anonymizovaná data. Zdroj: Vlastní zpracování.

- Ve třetím pohledu se párují poslední objednané množství „Quantity“ pro daný týden dodání „Prediction_week“, tedy množství s nejnižší týdenní diferencí „Difference_week“, s objednaným množstvím „Quantity“ a týdenní diferencí

„Difference_week“ vztažené ke stejnému datu dodání „Prediction_week“. Ve výsledné tabulce tedy existují sloupce obsahující hodnoty o ID dodavatele „Supplier_ID“, ID materiálu „Material_ID“, ID závodu „Plant_ID“, skutečně dodané množství deklarované v poslední předpovědi vztažené k požadovanému týdni dodání „bp1_Quantity“, předpovídané množství v týdnu, kdy byla vystavena zpráva s předpovědí „bp2_Quantity“, procentuální vyjádření přesnosti predikce „Quantity_difference“, kde pokud je předpověď větší než poslední predikované množství před dodáním materiálu, pak je hodnota větší než 100, difference termínu poslední predikce před požadovaným týdnem dodání materiálu a požadovaným týdnem dodání materiálu „bp1_Difference“, difference termínu předpovědi vztažená k týdnu, kdy bylo předpovídáno dodání materiálu a požadovaným týdnem dodání materiálu „bp2_Difference“ a týden vystavení zprávy s předpovědí „Prediction_week“. Výsledkem operace je virtuální tabulka obsahující jeden záznam pro každou kombinaci dat z týdne dodání a týdne předpovědi spolu se skutečně dodaným množstvím a původně objednaným množstvím, vztažených na stejný termín dodání. Z obou množství se následně vypočte nárůst nebo pokles. Vzniklá virtuální tabulka obsahuje údaje v celkem 43070 řádcích pro druhý balík zpráv.

	Supplier_ID	Material_ID	Plant_...	bp1_Quan...	bp2_Quan...	Quantity_difere...	bp1_Difference_w...	bp2_Difference_w...	Prediction_we...
1	1000872802	1038270329	6134	528	528	100.000000	0	0	2014-01-13
2	1000872802	1038270329	6134	528	528	100.000000	0	2	2014-01-13
3	1000872802	1038270329	6134	528	528	100.000000	0	4	2014-01-13
4	1000872802	1038270329	6134	528	528	100.000000	0	5	2014-01-13
5	1000872802	1038270329	6134	528	528	100.000000	0	6	2014-01-13
6	1000872802	1038270329	6134	264	264	100.000000	0	0	2014-01-27
7	1000872802	1038270329	6134	264	264	100.000000	0	1	2014-01-27

Query executed successfully. MICHAL-NTB\BP (12.0 RTM) | MICHAL-NTB\Michal (59) | master | 00:00:03 | 43070 rows

Obrázek 19: SELECT virtuálního pohledu číslo 3 obsahujícího anonymizovaná data. Zdroj: Vlastní zpracování.

- Ve čtvrtém pohledu jsou sečtené odchylky v předpovědích vyděleny počtem těchto sečtených odchylek „Avg_quantity_diff“, aby se podařilo získat celkovou odchylku stanovenou na cílový termín dodání. Je zde nastaven formát data předpovědi „Prediction_week“ tak, aby byl čitelný programem Microsoft Excel, a je zde vytvořen sloupec obsahující týdenní difference mezi poslední předpovědí na daný termín dodání a konkrétní předpovědí dodání materiálu „Difference_week“, podle které lze filtrovat výsledek a získat tak podklady k tvorbě grafu. Výsledná virtuální tabulka obsahuje pro

druhý balík dat přesně 2091 záznamů celkem pro všechny týdenní difference a všechny týdny požadovaného dodání materiálu. Jedná se o tabulku, která byla popsána před popisem činnosti jednotlivých logických částí, příkazů prováděných nad databázovou tabulkou obsahující nahromaděná data.

	Prediction_week	Avg_quantity_diff	Difference_week
1	2013-11-11 00:00:00.000	100.000000	0
2	2013-11-18 00:00:00.000	100.000000	0
3	2013-11-25 00:00:00.000	100.000000	0
4	2013-12-02 00:00:00.000	100.000000	0
5	2013-12-09 00:00:00.000	100.000000	0
6	2013-12-16 00:00:00.000	100.000000	0
7	2014-01-06 00:00:00.000	100.000000	0

Query executed successfully. | MICHAL-NTB\BP (12.0 RTM) | MICHAL-NTB\Michal (61) | master | 00:00:03 | 2091 rows

Obrázek 20: *SELECT* virtuálního pohledu číslo 4 obsahujícího anonymizovaná data. Zdroj: vlastní zpracování.

K předchozím operacím nad tabulkou s daty se nabízela i možnost využít možnosti relačních databází, jako je například příkaz *CURSOR*, nicméně od tohoto řešení bylo upuštěno jednak z důvodu vyšší náročnosti na znalost relačních databází, a jednak také proto, že kvůli komplexnosti řešení autor práce od počátku přesně nevěděl, jak data pomocí databáze vhodně zpracovat. Postupným definováním jednotlivých pohledů měl autor práce lepší přehled o tom, co se děje s daty, a byla tak minimalizována možnost chyb.

3.4 Operace v programu Microsoft Excel

Po přetransformování dat do podoby tabulky o třech sloupcích, je celá tato virtuální tabulka přenesena do tabulky v programu Microsoft Excel. Jedná se o sloupce obsahující požadovaný týden dodání materiálu, poměr předpovídaného množství a skutečně dodaného množství, difference (týdenní) mezi termínem dodání a termínem předpovědi na daný termín. Práce s již zmiňovanými „čtyřtýdenními“ není v databázi ani v Microsoft Excelu implementována z důvodu nutnosti vytvoření nových pohledových tabulek pro tento formát dat, a z důvodu zamýšlené změny z formátu data v týdnech na formát dní a měsíců.

Pro tvorbu grafu je velmi důležité, aby byla data v prvním sloupci „Prediction_week“ ve formátu data podporovaného programem Microsoft Excel. Autor práce dospěl k závěru, že vhodným formátem je formát *datetime* podporovaný databázovým systémem Microsoft

SQL Server 2014 a jazykem SQL. V případě, že by datum nebyl ve správném formátu, který je podporován i programem Microsoft Excel, data by nebyla na ose znázorněna se správnými hodnotami.

3.4.1 První import tabulky

Převedení virtuální tabulky vytvořené čtvrtým pohledem do listu programu Microsoft Excel je velmi jednoduché. Tabulku lze v anglické verzi programu nainportovat skrze záložku „Data“, následně je potřeba kliknout na záložku „From Other Sources“, protože se jedná o tabulku v databázi Microsoft SQL Server. Po poklepu se rozbálí nabídka a hned první záložka odkazuje na SQL Server. Klikneme na tuto záložku, zadáme název serveru, tedy „localhost“. Protože databáze se nachází na stejném stroji jako Excelový soubor, jde tedy o lokální server. Jako autentifikační metodu vybereme tu, kterou využíváme. V rámci této bakalářské práce je využita jednoduchá Windows autentifikace. Když se nám podaří připojit databázi, zadáme nebo zvolíme požadovanou tabulku nebo pohled k importu a necháme ji nainportovat na zvolené místo do listu.

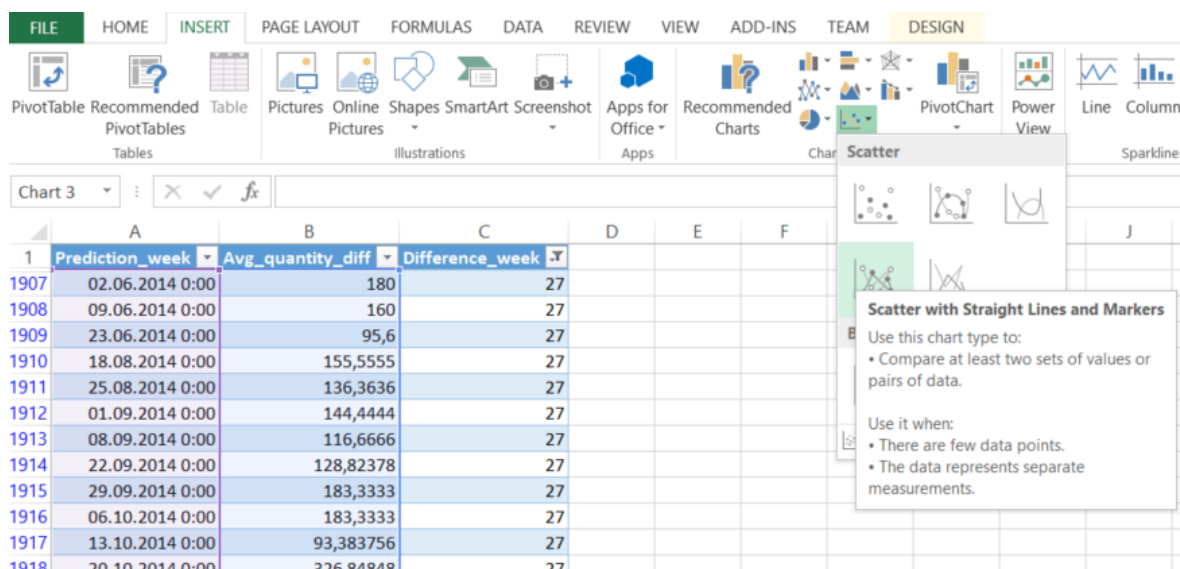
3.4.2 Opakovaný import tabulky

V případě, že import provádíme opakovaně, lze využít jiné, kratší cesty. Znovu klikneme na záložku „Data“, následně zvolíme „Existing connections“, a zde uvidíme výpis existujících spojení a předchozích importovaných tabulek. Zde vybereme zvolenou tabulku „view4“ a nainportujeme. Je podstatné zmínit, že data v importované tabulce jsou vždy ta, která jsou uložena v Microsoft SQL Serveru. Pokud tedy data v databázi změníme, nejedná se o předchozí importovanou verzi, ale o verzi aktuální.

Po zdařilém importu dat do listu Excelu vidíme tabulku obsahující údaje pro všechny týdny dodání a všechny týdenní diference. Tabulka již obsahuje filtry, takže podle toho, jaký graf a s jakým rozdílem mezi předpovědí a datem doručení materiálu chceme tvořit, zvolíme filtrem jednu týdenní diferenci a necháme si ji vyfiltrovat. Vytvořená vyfiltrovaná tabulka obsahuje již pouze hodnoty pro tvorbu grafu se zvolenou diferencí. Pokud vybereme data, můžeme přistoupit k tvorbě grafu.

3.4.3 Jak vytvořit graf

Výstupní graf vytvoříme tak, že vybereme data ze sloupce „Prediction_week“ (týden, na kdy se vztahuje předpověď), a ze sloupce „Avg_quantity_diff“ (průměrný procentuální rozdíl mezi množstvím z týdne předpovědi, a z posledních předpovědí odeslaných těsně před doručením materiálu). Po vybrání dat klikneme v záložce „Insert“ (Vložit), podzáložce „Charts“ (Grafy), na záložku „Insert Scatter Chart“ (vložit bodový graf). Zde vybereme typ grafu, který chceme vykreslit a kliknutím jej necháme zobrazit.

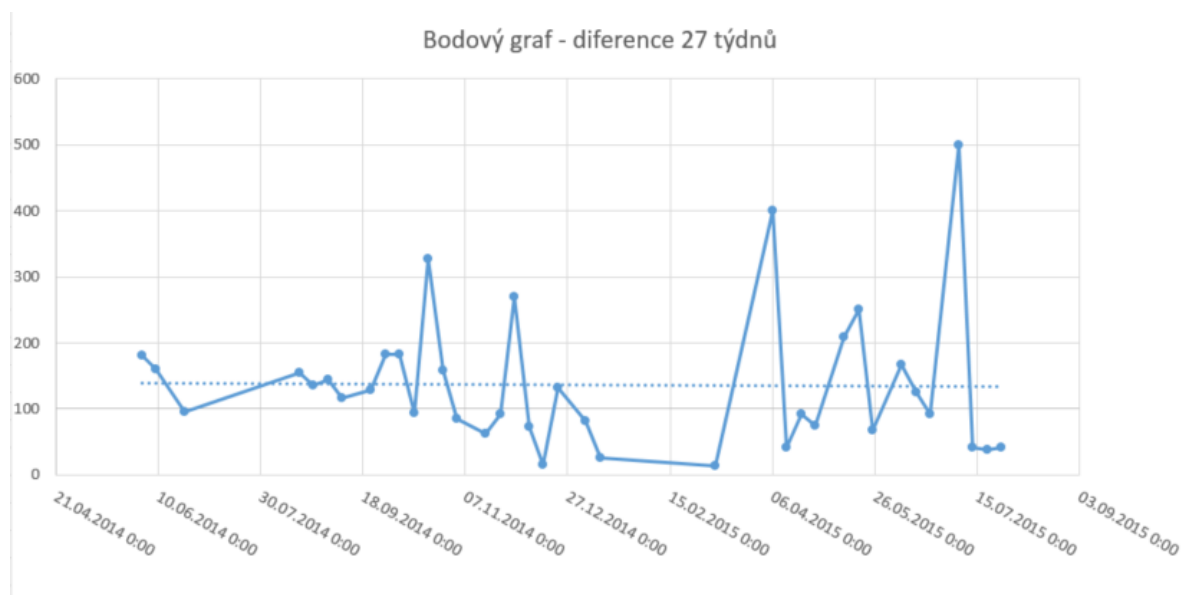


Obrázek 21: Tvorba bodového grafu týdenní diferencí 27 týdnů. Zdroj: Vlastní zpracování.

Výstupní graf znázorňuje přesnost předpovědi na určitý týden. Přesnost je znázorněna na svislé ose a jednotlivá data, ke kterým je přesnost vztažena, jsou vypsána na vodorovné ose. Hodnoty přesnosti jsou udávány v procentech a v případě, že odběratel (který je odesílatelem) odvolávek nadhodnotí v předpovědi potřebné množství, je hodnota v grafu vyšší než 100. Analogicky, pokud odběratel podhodnotí předpovídané potřebné množství na daný termín, bude hodnota v grafu nižší než 100.

Graf můžeme dále upravovat například přidáním „Trendline“ (spojnice trendu) dle našeho výběru, díky které můžeme zhruba vidět, jak se vyvíjí přesnost v čase, nebo upravením počátečních hodnot na osách tak, aby bylo výstupní vodorovnou hodnotou 100 %, což je ukazatel nezměněného objemu požadovaného množství materiálu, tudíž ideální případ.

Na následujícím obrázku si může čtenář prohlédnout graf znázorňující přesnost predikcí vztažených k týdenní diferenci mezi předpovědí a datem dodání rovné 27. V grafu je přidána lineární trendová funkce. Na vodorovné ose je vyznačen cílový datum dodání, na svislé ose jsou znázorněna procenta přesnosti predikce. Výchozí hodnota procentuální změny je v grafu nastavena na 100 %.



Obrázek 22: Graf přesností predikcí s týdenní diferencí rovné 27. Zdroj: Vlastní zpracování.

3.5 Trendy v logistice

Poslední dobou je od managementu firem čím dál více požadavků na možnost sledovat data z výroby, logistiky a dalších disciplín. Dnešní informační systémy se proto snaží nabídnout co nejvíce možných ukazatelů, které zjednoduší manažerům rozhodování v klíčových okamžicích. Pomohou tak společnosti dosáhnout lepších výsledků, a získat tak před svými konkurenty konkurenční výhodu. Trendem je tedy sledovat všechna data, která plynou z výroby a provádět nad nimi, pokud možno „real-time“ analýzu (analýzu v reálném čase).

Jedním z dalších trendů současné logistiky je minimalizace prostojů ve výrobě. Auto-motive průmysl je jedním z nejvíce prosperujících průmyslů vůbec. Díky velikosti si automobilky mohou dovolit diktovat svým dodavatelům podmínky, a proto jsou také automobilky na pomyslném vrcholu technologického pokroku – využívají totiž nejvíc technologií, které se postupně zpřístupňují i menším společnostem. Minimalizace nákladů na výrobu, tedy

analogicky i minimalizace prostojů výroby, vede k menším nákladům a menší výrobní ceně produktu. Při stejné ceně tedy roste marže automobilky, což je výhodné pro její akcionáře. Z opačného pohledu je možné o snížené náklady snížit cenu produktu, což vede k většímu zájmu zákazníků – kupců automobilů a jejich náhradních dílů.

S minimalizací prostojů souvisí i dobře plánované zásoby, což je ve výrobě a logistice dalším velmi sledovaným tématem. Společnosti se proto snaží, aby co nejlépe uměly plánovat svoje zásoby, tedy aby mohly mít menší skladovou plochu, menší počet skladníků, méně vytápěných prostor, zpracovávaly kvalitní suroviny, kterým ještě nevypršela doba trvanlivosti a podobně.

Praktickým cílem práce proto bylo vymyslet řešení, které pomůže tyto trendy sledovat, naplňovat a bude použitelné v praxi.

3.5.1 Využitelnost poznatků z práce v praxi

Sestrojený graf, který lze s trochou úprav procesů a kódu plně automatizovat a implementovat do některého z produktů společnosti AIMTEC. Graf by měl sloužit k orientačnímu posouzení přesnosti předpovědí potřeb odběratele napříč více jeho dodavateli. Těmto dodavatelům, kteří odesílají data společnému odběrateli, by mohlo být v rámci jednoho z produktů společnosti AIMTEC umožněno sledovat vývoj grafu v minulosti, případně i předpokládaný vývoj v budoucnosti, a zlepšit tak procesy ve své firmě.

Použití poznatků z grafického znázornění přesnosti předpovědí jednoho odběratele napříč jeho dodavateli mohou tito dodavatelé využít například k tomu, aby lépe plánovali skladové zásoby. Toto vede ke snížení nákladů na skladování, protože dodavatel bude mít informaci o tom, jak se vyvíjí přesnost objednávek jeho produktů, a bude tak moci lépe odhadnout množství jeho produktů, které bude jeho odběratelem poptáváno, a také čas jeho potřeby. Zjištěné poznatky je možné nabídnout i odběrateli, který může tato data zanalyzovat a upravit tak své předpovědi s cílem zvýšení jejich dlouhodobé přesnosti.

Dalším podstatným benefitem pro dodavatele sledujícího výstupní graf může být menší nejistota a zvýšení akceschopnosti v dodávkovém řetězci zdrojů pro jeho výrobu. Bude moci

lépe naplánovat, kolik kterého materiálu bude pro svoji výrobu potřebovat, tudíž nedojde k problému nedostatku zásob pro výrobu z důvodu větší poptávky, než bylo plánováno, nebo se toto riziko alespoň minimalizuje.

Díky informacím získaným z grafu přesnosti předpovědí odběratele bude tedy dodavatelova výroba, skladování, plánování a třeba i doprava efektivnější, a to znamená, že ušetří náklady. Dodavatel dosáhne taktéž minimalizace výskytu – nebo úplné eliminace situací, kdy by musel platit odběrateli pokutu za zastavení nebo zpomalení výroby. Je totiž poměrně častou praxí, že velké automobilky mají s dodavateli smluvní vztah, který stanovuje, že pokud nefungují dodávky materiálu, jsou dodavateli vyměřeny vysoké pokuty, které vynahrazují ušlý zisk prodeje výrobků odběratele. Tyto pokuty jsou pro dodavatele mnohdy devastující, proto je přinejmenším vhodné se pokusit je jakkoliv eliminovat.

Závěr

V práci byly pojmenovány trendy z oblasti logistiky a výroby, které souvisí s elektronickou výměnou dat. Příkladem identifikovaného trendu je potřeba minimalizace skladových zásob nebo potřeba dlouhodobého monitoringu požadavků odběratele na dodání materiálu jeho dodavateli. V práci byl naplněn tento hlavní cíl i všechny dílčí cíle.

V rámci kapitoly Teoretická východiska práce bylo definováno, co je to EDI a byla popsána struktura EDI zpráv. Dále byly vysvětleny důležité termíny, které je potřeba znát, aby se čtenář zorientoval v terminologii informačních systémů. Pro potřeby naplnění dalších dílčích cílů práce byly vysvětleny základy programování a práce s databází.

Byl vytvořen parser, pomocí kterého byla extrahována data z vybraných anonymizovaných EDI zpráv VDA4905, získaných od společnosti AIMTEC. Byl navržen způsob uložení dat v databázi a nad těmito daty byla provedena analýza, jejímž výstupem je jednoduchá tabulka sloužící jako základ k tvorbě grafu, který byl zvolen jako vhodná vizualizace výstupních dat. Výstupní data znázorňují přesnost předpovědí množství materiálu, které je požadováno ve výrobě odběratele napříč jeho dodavateli. V práci byly navrženy možnosti využití výstupního grafu v rozhodování managementu ve výrobě a logistice. Vyvinuté řešení lze s menšími úpravami implementovat do informačních systémů společnosti AIMTEC.

Autor práce se díky studiu teoretických podkladů souvisejících s prací na tomto textu naučil, jak efektivně třídit informace získané z mnoha pramenů. Díky možnosti použití anonymizovaných dat poskytnutých společnostmi AIMTEC se autor dozvěděl, jak funguje elektronická výměna dat, a naučil se číst zprávy z této komunikace. Díky konzultacím a samostudiu problematiky autor vyvinul funkční řešení, pomocí kterého lze několika kroky generovat graf, který může posloužit k další analýze.

Popisované řešení má reálný potenciál ve výrobě a logistice. Management společností dodávajících svoje produkty do jedné z předních světových automobilek se v případě implementace řešení do aplikací business intelligence může opřít o přínosné informace, které pomohou učinit správná rozhodnutí, ušetřit tak náklady a vygenerovat zisk.

Seznam obrázků

OBRÁZEK 1: ÚSPORA PLYNOUCÍ Z ELEKTRONIZACE FAKTURACE ZE STRANY ODESÍLATELE	13
OBRÁZEK 2: ÚSPORA PLYNOUCÍ Z ELEKTRONIZACE FAKTURACE ZE STRANY PŘÍJEMCE.....	13
OBRÁZEK 3: CENNÍK VYTVOŘENÝ ZA POMOCÍ XML KÓDU	14
OBRÁZEK 4: DEKLARACE TŘÍDY V JAZYKU JAVA	26
OBRÁZEK 5: PODMÍNKA IF, ELSE IF, ELSE	27
OBRÁZEK 6: CYKLUS FOR.....	28
OBRÁZEK 7: CYKLUS WHILE.....	28
OBRÁZEK 8: CYKLUS DO-WHILE.....	28
OBRÁZEK 9: PŘÍKAZ K VYTVOŘENÍ TABULKY.....	30
OBRÁZEK 10: VYMAZÁNÍ TABULKY	30
OBRÁZEK 11: PŘÍKAZ SELECT.....	31
OBRÁZEK 12: PŘÍKAZ K VYTVOŘENÍ VIRTUÁLNÍ TABULKY.....	31
OBRÁZEK 13: DIAGRAM PROCESŮ	37
OBRÁZEK 14: PŘÍKAZ NA EXTRAKTCI SOUBORŮ ZE SLOŽEK (PŘÍKAZOVÁ ŘÁDKA)	39
OBRÁZEK 15: KONTROLNÍ VÝPIS V KONZOLI PO SKONČENÍ PARSERU A NAPLNĚNÍ DAT DO DATABÁZE.....	46
OBRÁZEK 16: SELECT S VÝTAHEM SLOUPCŮ, VIRTUÁLNÍHO POHLEDU ČÍSLO 0 OBSAHUJÍCÍHO ANONYMIZOVANÁ DATA	47
OBRÁZEK 17: SELECT VIRTUÁLNÍHO POHLEDU ČÍSLO 1 OBSAHUJÍCÍHO ANONYMIZOVANÁ DATA	48
OBRÁZEK 18: SELECT VIRTUÁLNÍHO POHLEDU ČÍSLO 2 OBSAHUJÍCÍHO ANONYMIZOVANÁ DATA	48
OBRÁZEK 19: SELECT VIRTUÁLNÍHO POHLEDU ČÍSLO 3 OBSAHUJÍCÍHO ANONYMIZOVANÁ DATA	49
OBRÁZEK 20: SELECT VIRTUÁLNÍHO POHLEDU ČÍSLO 4 OBSAHUJÍCÍHO ANONYMIZOVANÁ DATA	50
OBRÁZEK 21: TVORBA BODOVÉHO GRAFU TÝDENNÍ DIFERENCÍ 27 TÝDNŮ	52
OBRÁZEK 22: GRAF PŘESNOSTÍ PREDIKCÍ S TÝDENNÍ DIFERENCÍ ROVNÉ 27.	53

Seznam tabulek

TABULKA 1: POPIS ELEMENTŮ PRVNÍ ÚROVNĚ VE STANDARDU UN/EDIFACT	18
TABULKA 2: POPIS ELEMENTŮ DRUHÉ ÚROVNĚ VE STANDARDU UN/EDIFACT	18
TABULKA 3: ZPRÁVA VDA 4905 – STRUČNÝ POPIS JEDNOTLIVÝCH SEGMENTŮ	21
TABULKA 4: ZPRÁVA VDA 4905 – ODVOLÁVKA, POPIS SEGNETU „511“	22
TABULKA 5: ZPRÁVA VDA 4905 – ODVOLÁVKA, POPIS SEGMENTU „512“	22
TABULKA 6: ZPRÁVA VDA 4905 – ODVOLÁVKA, POPIS SEGMENTU „513“	23
TABULKA 7: ZPRÁVA VDA 4905 – ODVOLÁVKA, POPIS SEGMENTU „514“	24
TABULKA 8: ZPRÁVA VDA 4905 – ODVOLÁVKA, POPIS SEGMENTU „519“	24
TABULKA 9: ZPRÁVA VDA 4905 – ODVOLÁVKA, KÓDOVÁ OZNAČENÍ.....	25
TABULKA 10: DATOVÉ TYPY V PROGRAMOVACÍM JAZYKU JAVA	27
TABULKA 11: TABULKA S DATY, KTERÁ JSOU EXTRAHOVÁNA PARSEREM.....	41
TABULKA 12: TABULKA OBSAHUJE PŘÍKLAD VÝSTUPNÍCH DAT V JEDNOTLIVÝCH SLOUPCÍCH (ZDE ŘÁDCÍCH) .	45

Seznam použité literatury

- [1] VYMĚTAL, Dominik. *Informační systémy v podnicích: teorie a praxe projektování*. 1. vyd. Praha: Grada, 2009. Průvodce (Grada). ISBN 978-80-247-3046-2.
- [2] Information system. *APICS* [online]. [cit. 2015-12-26]. Dostupné z: <http://www.apics.org/dictionary/dictionary-information?ID=2012.0>
- [3] BASL, Josef. *Podnikové informační systémy: podnik v informační společnosti*. 2., výrazně přeprac. a rozš. vyd. Praha: Grada, 2008, 283 s. Management v informační společnosti. ISBN 978-80-247-2279-5.
- [4] Enterprise Resource Planning. *Investopedia* [online]. [cit. 2015-12-26]. Dostupné z: <http://www.investopedia.com/terms/e/erp.asp>
- [5] Co je EDI? *EdiZone* [online]. [cit. 2015-12-28]. Dostupné z: <http://www.edizone.cz/elektronicka-vymena-dat-edi/co-je-edi/>
- [6] KOLEKTIV AUTORŮ. *Elektronický obchod a EDI*. 1. vyd. Brno: UNIS Publishing, 1996, 216 s.
- [7] EDI Definition, *TechTerms* [online]. [cit. 2015-12-28]. Dostupné z: <http://techterms.com/definition/edi>
- [8] Still massive cost savings in moving to e-invoicing: €6.40 per invoice for invoice issuers & €10. *Jack Large's CMT File* [online]. [cit. 2015-12-26]. Dostupné z: <https://ctmfile.com/story/still-massive-cost-savings-in-moving-to-e-invoicing-6.40-per-invoice-for-in#.VoFbpejhChd>
- [9] KOSEK, Jiří. *XML pro každého: podrobný průvodce*. 1. vyd. Praha: Grada, 2000. ISBN 80-716-9860-1.
- [10] EDI vs. XML. *EDiDev* [online]. [cit. 2015-12-29]. Dostupné z: http://www.edidev.com/articles/XmUvsEdi/Xml_vs_Edi.html

- [11] About ebXML. *EbXML - Enabling a global electronic market* [online]. [cit. 2016-04-10]. Dostupné z: <http://www.ebxml.org/geninfo.htm>
- [12] D97A DELFOR. *EDIFACTORY* [online]. [cit. 2015-12-30]. Dostupné z: <http://www.edifactory.de/msglist.D97A?m=DELFOR>
- [13] PART 4 UNITED NATIONS RULES FOR ELECTRONIC DATA INTERCHANGE FOR ADMINISTRATION, COMMERCE AND TRANSPORT. *UN/EDIFACT DRAFT DIRECTORY: UN/EDIFACT Syntax Implementation Guidelines* [online]. [cit. 2015-12-30]. Dostupné z: <https://www.unece.org/fileadmin/DAM/trade/untdid/texts/d423.htm#p9.2>
- [14] DESADV. *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport* [online]. [cit. 2015-12-30]. Dostupné z: http://www.unece.org/trade/untdid/d00a/trmd/desadv_c.htm
- [15] ORDERS. *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport* [online]. [cit. 2015-12-30]. Dostupné z: http://www.unece.org/trade/untdid/d00a/trmd/orders_c.htm
- [16] ORDRSP. *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport* [online]. [cit. 2015-12-30]. Dostupné z: http://www.unece.org/trade/untdid/d00a/trmd/ordrsp_c.htm
- [17] INVOIC. *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport* [online]. [cit. 2015-12-30]. Dostupné z: http://www.unece.org/trade/untdid/d00a/trmd/invoic_c.htm
- [18] INVRPT. *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport* [online]. [cit. 2015-12-30]. Dostupné z: http://www.unece.org/trade/untdid/d00a/trmd/invrpt_c.htm
- [19] APERAK. *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport* [online]. [cit. 2015-12-30]. Dostupné z: http://www.unece.org/trade/untdid/d00a/trmd/aperak_c.htm

- [20] [Edifact. *EDIBASICS* [online]. [cit. 2015-12-30]. Dostupné z: <http://www.edibasics.com/edi-resources/document-standards/edifact/>
- [21] Delivery instruction VDA 4905/1. *VW EDI Implementation Guidelines* [online]. [cit. 2015-12-30]. Dostupné z: http://edi.skoda-auto.cz/soubor/VDA4905_e.pdf
- [22] HEROUT, Pavel. *Učebnice jazyka Java. 5., rozš. vyd.* České Budějovice: Kopp, 2010. ISBN 978-80-7232-398-2.
- [23] SINGH, S.K. *Database systems concepts, design and applications. 2.* Delhi: Dorling Kindersley (India), 2011. ISBN 9788131760925.
- [24] What is database?: database definition. *TechTarget* [online]. [cit. 2016-04-15]. Dostupné z: <http://searchsqlserver.techtarget.com/definition/database>
- [25] Download Microsoft JDBC Drivers 6.0(Preview), 4.2, 4.1, and 4.0, for SQL Server from Official Microsoft Download Center. *Microsoft* [online]. [cit. 2016-04-15]. Dostupné z: <https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11774>

Seznam příloh

Příloha A: Struktura zprávy DELFOR ve formátu UN/EDIFACT. Zdroj: Vlastní zpracování, [12].

Příloha B: Ukázková zpráva VDA 4905 obsahující dva druhy materiálu. Zdroj: Data společnosti AIMTEC.

Příloha C1: Syntaxe SQL příkazu view0

Příloha C2: Syntaxe SQL příkazu view1

Příloha C3: Syntaxe SQL příkazu view2

Příloha C4: Syntaxe SQL příkazu view3

Příloha C5: Syntaxe SQL příkazu view4

Příloha D: CD disk (parser Extractor.java, Javadoc dokumentace k parseru, JDBC ovladač, SQL příkaz view0, SQL příkaz view1, SQL příkaz view2, SQL příkaz view3, SQL příkaz view4)

Přílohy

Příloha A:

Seg.	Popis	M/C	Zanoření
UNH	Servisní segment - Hlavička dokumentu	M	
BGM	Začátek zprávy	M	
DTM	Datum/čas	M	
FTX	Doplňující text	C	
Skupina segmentů 1 - údaje o kontraktu		C	
RFF	Reference	M	
DTM	Datum/čas	C	
Skupina segmentů 2 - údaje o odesílateli		C	
NAD	Název a adresa Odesílatele	M	
Skupina segmentů 3 - reference k danému odesílateli		C	
RFF	Reference	M	
DTM	Datum/čas	C	
Skupina segmentů 4 - identifikace odpovědné osoby odes.		C	
CTA	Informace o kontaktní osobě	M	
COM	Kontakt	C	
Skupina segmentů 5 - transportní údaje		C	
TDT	Detaily transportu	M	
DTM	Datum/čas	C	
Skupina segmentů 6 - Detail o dodávce		C	
GIS	Hlavní indikátor	M	
Skupina segmentů 7 - data o místě příjemce		C	
NAD	Název a adresa	M	
LOC	Lokace - brána, přístav..	C	
FTX	Doplňující text	C	
Skupina segmentů 8 - reference k příjemci		C	
RFF	Reference	M	
DTM	Datum/čas	C	
Skupina segmentů 9 - dokumenty vztahující se k příjemci		C	
DOC	Údaje k dokumentu	M	
DTM	Datum/čas	C	
Skupina segmentů 10 - identifikace osoby příjemce		C	
CTA	Informace o kontaktní osobě	M	
COM	Kontakt	C	
Skupina segmentů 11 - transportní údaje		C	
TDT	Detaily transportu	M	
DTM	Datum/čas	C	
Skupina segmentů 12 - informace o produktu		C	

LIN	Informace o protuktu	M	
PIA	Dodatečné produkt ID	C	
IMD	Popis produktu - ne ID	C	
MEA	Informace o balení - měrné jednotky	C	
ALI	Původ a jiné dodatečné informace	C	
GIN	Identifikační čísla produktu	C	
GIR	Další identifikační čísla	C	
LOC	Požadované místo umístění	C	
DTM	Datum/čas	C	
FTX	Doplňující text	C	
Skupina segmentů 13 - referenční údaje o položce		C	
RFF	Reference	M	
DTM	Datum/čas	C	
Skupina segmentů 14 - specifika nákladu		C	
TDT	Detaily transportu	M	
DTM	Datum/čas	C	
Skupina segmentů 15 - specifikace množství a data		C	
QTY	Množství	M	
DTM	Datum	C	
Skupina segmentů 16 - referenční údaje		C	
RFF	Reference - často avízo	M	
DTM	Datum/čas	C	
Skupina segmentů 17 - další informace o produktu		C	
SCC	Týdenní/denní - ostatní podmínky	M	
Skupina segmentů 18 - specifikace množství a data		C	
QTY	Množství	M	
DTM	Datum/čas	C	
Skupina segmentů 19 - reference k předchozímu segmentu		C	
RFF	Reference	M	
DTM	Datum/čas	C	
Skupina segmentů 20 - balící informace		C	
PAC	Typ přepravní jednotky	M	
MEA	Rozměry balení	C	
QTY	Množství balení	C	
DTM	Datum/čas	C	
Skupina segmentů 21 - identifikace balení		C	
PCI	Identifikační čísla balení	M	
GIN	ID jednotek produktů	C	
Skupina segmentů 22 - údaje o místě dodání		C	
NAD	Název a adresa	M	
LOC	Lokace - brána, přístav..	C	
FTX	Doplňující text	C	
Skupina segmentů 23 - Informace a dokumenty		C	


```

    , [Supplier_ID]
    , [Material_ID]
    , [Discharge_place_ID]
    , [Plant_ID]
    , [Order_number]
    , [Requirement_code]
    , [Message_date]
    , [Message_week]
    , [Prediction_week]
    , [Quantity]
    , [Message_month]
    , [Prediction_month]
    , [Difference_week]
    , [Difference_month]
    , [Date_type]
    , [Prediction_date]
    , [Original_prediction_date]
    , COUNT(1) as Lines_count
FROM [BP].[dbo].[BP_table]
GROUP BY
    [Message_cust_ID]
    , [Supplier_ID]
    , [Material_ID]
    , [Discharge_place_ID]
    , [Plant_ID]
    , [Order_number]
    , [Requirement_code]
    , [Message_date]
    , [Message_week]
    , [Prediction_week]
    , [Quantity]
    , [Message_month]
    , [Prediction_month]
    , [Difference_week]
    , [Difference_month]
    , [Date_type]
    , [Prediction_date]
    , [Original_prediction_date];
GO

```

Příloha C2:

```

DROP VIEW [view1]
GO

```

```

CREATE VIEW [view1] AS SELECT *
FROM [view0] [in_tbl]
WHERE
    [in_tbl].[Message_Cust_ID] = (
SELECT MAX([Message_Cust_ID])
FROM [view0] [in1_tbl]
WHERE [in1_tbl].[Supplier_ID] = [in_tbl].[Supplier_ID]
and [in1_tbl].[Material_ID] = [in_tbl].[Material_ID]
and [in1_tbl].[Prediction_date] = [in_tbl].[Prediction_date]
and [in1_tbl].[Date_type] = [in_tbl].[Date_type]
and [in1_tbl].[Difference_week] = [in_tbl].[Difference_week]
and [in1_tbl].[Plant_ID] = [in_tbl].[Plant_ID]);

```

Příloha C3:

```
DROP VIEW [view2]
GO

CREATE VIEW [view2] AS
SELECT
    [Supplier_ID]
    , [Material_ID]
    , [Difference_week]
    , [Prediction_week]
    , [Plant_ID]
    , sum([Quantity]) AS [Quantity]
    , COUNT(*) AS [Record_count]
FROM [view1]
GROUP BY
    [Supplier_ID]
    , [Material_ID]
    , [Difference_week]
    , [Prediction_week]
    , [Plant_ID];
GO
```

Příloha C4:

```
DROP VIEW [view3]
GO

CREATE VIEW [view3] AS
SELECT
    [bp1].[Supplier_ID]
    , [bp1].[Material_ID]
    , [bp1].[Plant_ID]
    , [bp1].[Quantity] AS [bp1_Quantity]
    , [bp2].[Quantity] AS [bp2_Quantity]
    , COALESCE(( [bp2].[Quantity] / NULLIF([bp1].[Quantity],0) ) * 100, 100) AS
[Quantity_difference]
    , [bp1].[Difference_week] AS [bp1_Difference_week]
    , [bp2].[Difference_week] AS [bp2_Difference_week]
    , [bp1].[Prediction_week]
FROM
    [view2] [bp1] join [view2] [bp2] ON (
    [bp1].[Supplier_ID] = [bp2].[Supplier_ID]
    and [bp1].[Material_ID] = [bp2].[Material_ID]
    and [bp1].[Plant_ID] = [bp2].[Plant_ID]
    and [bp1].[Prediction_week] = [bp2].[Prediction_week]
    )
WHERE [bp1].[Difference_week] = (
SELECT
    MIN([Difference_week])
FROM [view2]
WHERE
    [bp1].[Supplier_ID] = [Supplier_ID]
```

```
and [bp1].[Material_ID] = [Material_ID]
and [bp1].[Plant_ID] = [Plant_ID];
```

Příloha C5:

```
DROP VIEW [view4]
GO
```

```
CREATE VIEW [view4] as
SELECT
    CAST([Prediction_week] AS datetime) AS [Prediction_week]
    ,SUM([Quantity_difference])/COUNT(*) AS [Avg_quantity_diff]
    ,[bp2_Difference_week] AS [Difference_week]
FROM
    [view3]
GROUP BY
    [Prediction_week], [bp2_Difference_week];
```

Abstrakt

DRDA, M. Sledování trendů na základě výměny elektronických dokumentů (EDI).
Bakalářská práce. Plzeň: Fakulta ekonomická ZČU v Plzni, 70 s., 2016

Klíčová slova: EDI, informační systémy, předpověď, odvolávka, trend, parser, zpracování dat

Prezentovaná bakalářská práce se zabývá identifikací trendů, které lze sledovat v elektronické výměně dat. Hlavním cílem této práce bylo najít tyto trendy a popsat je.

V teoretické části bylo vysvětleno, co je to EDI a jaké známe typy zpráv používaných v elektronické výměně dat. Dále bylo definováno, co je to informační systém a jaké jsou alternativy EDI. Z důvodu použití programovacích a dotazovacích jazyků v praktické části byly popsány základy jazyka Java a SQL.

V praktické části byly splněny cíle týkající se extrakce dat z vybraných EDI zpráv. Bylo navrženo vhodné uložení a předveden způsob využití informací získaných z použitých zpráv.

V závěru práce jsou shrnuty trendy, které v poslední době určují vývoj logistiky a výroby. V práci je také popsáno, jak je možné využít poznatky z práce v praxi.

Abstract

DRDA, M. Trending based on the exchange of electronic documents (EDI). Bachelor thesis. Pilsen: Faculty of Economics, University of West Bohemia in Pilsen, 70 p., 2016

Key words: EDI, information systems, prediction, call-off, trend, parser, data processing

The presented Bachelor thesis deals with the identification of trends, which can be followed in the electronic data interchange. The main goal of this thesis was to search for these trends and to describe them.

In the theoretical part it is explained, what is the meaning of EDI and what kinds of messages are used in the electronic data interchange. Also it was defined, what is the information system and what the alternatives of EDI are. Because of the usage of programming and query languages in the practical part, the basics of Java and SQL languages were explained.

In the practical part, the goal was fulfilled with the extraction of data from a batch of selected EDI messages and a suitable way of storing information parsed from the used messages was suggested and performed.

At the end of the thesis recent trends which drive the logistics and production were summarized and it was suggested how to use the knowledge from the thesis in the practice.