Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních techlologií

Diplomová práce

Implementace datového můstku mezi zdrojovým a cílovým IS pomocí Knime Analytics Platform

Bc. Pavel Bačík

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Pavel Bačík

Informatika

Název práce

Implementace datového můstku mezi zdrojovým a cílovým IS pomocí Knime Analytics Platform

Název anglicky

Data bridge implementation between source and destination IS by Knime Analytics Platform

---

Cíle práce

Diplomová práce je teoreticky i prakticky orientovaná na využití nástroje Knime Analytics Platform k řešení přenosu dat mezi zdrojovým a cílovým systémem. Hlavním cílem je na základě provedené literární rešerše analyzovat zvolené nástroje a využít je k realizaci ze zdrojového systému do cílové aplikace.

Metodika

Teoretická část diplomové práce je založena na analýze možností aplikace Knime Analytics Platform. V dalším kroku budu analyzovat možnosti načtení dat ze zdrojového systému, stejně tak i možnosti spojení s cílovým systémem.

Praktická část se zaměří na samotnou implementaci řešení přenosu dat mezi zdrojovým a cílovým systémem, analýze konkrétních řešení a porovnání jejich funkčnosti, rychlosti a využitelnosti v ostrém provozu.

Na základě poznatků z teoretické a praktické části bude následovat zhodnocení možností využití tohoto nástroje.

Doporučený rozsah práce

60 stran

Klíčová slova

Datový můstek, Knime, Import dat, Export dat, Formáty přesunu dat

---

Doporučené zdroje informací

BASL, J. – ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI, – BLAŽÍČEK, R. *Podnikové informační systémy : podnik v informační společnosti.* Praha: Grada, 2008. ISBN 978-80-247-2279-5.

POUR, J. – DOHNAL, J. *Architektury informačních systémů v průmyslových a obchodních podnicích.* Praha: Ekopress, 1997. ISBN 80-86119-02-5.

SILIPO, KNIME Beginner's Luck. Zurich: KNIME Press, 2018. ISBN 978-3-033-02850-0

VOŘÍŠEK, J. *Strategické řízení informačního systému a systémová integrace.* Praha: Management Press, 1997. ISBN 80-85943-40-9.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Edita Šilerová, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 11. 10. 2019

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 14. 10. 2019

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 12. 01. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Implementace datového můstku mezi zdrojovým a cílovým IS pomocí Knime Analytics Platform" jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 29.11.2022 

Poděkování

Rád(a) bych touto cestou poděkoval(a) paní doc. Ing. Editě Šilerové, Ph.D. za přijetí vlastního tématu práce a její vedení, dále Bc. Renému Baranovi za možnost vytvořit schéma, které v rámci diplomové práce není jen simulací možností propojení ale reálným řešením, které bude nasazeno.

Implementace datového můstku mezi zdrojovým a cílovým IS pomocí Knime Analytics Platform

Abstrakt

Tato diplomová práce je o implementaci Datového můstku v male společnosti, které potřebuje trošku vylepšit proces transformace dat z ekonomického systému do e-shop. Je to příležitost vyzkoušet nově nalezený software na www stránkách a zjistit, jestli dokáže nároky jako transformace dat ze souborů do databáze cílového systému. Dále bude zhodnocena tato záležitost z ohledů výdajů na vývoj a implementace takového softwaru.

Je samozřejmě míněno tento software reálně použít, nejedná se čistě o teoretický návrh Datového můstku v této práci.

Klíčová slova: Datový můstek, Knime, Import dat, Export dat, Formáty přesunu dat

Data bridge implementation between source and destination IS by Knime Analytics Platform

Abstract

This thesis is about implementation of Data bridge into small company that needs improvement in matter of transforming data from economy system do it's e-shop. It is chance to try out new found software on www pages, see if it's capable of handling such feat as transforming data from files to database of target system. Also it will be evaluated in matter of finances and extra expenses needed to implement such software.

This software is also meant to be truly used, it is not about creating Data bridge in theory.

Keywords: data bridge, Knime, data import and export, data formats

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Tato diplomová práce vzniká ve spolupráci se společností satelity-ul.cz (dále se budu odkazovat jako majitele e-shopu), který je dodavatel satelitních technologií s kamennou prodejnou v Ústí nad Labem. Této firmě dodává IT služby pan Bc. René Baran (dále jen programátor e-shopu), jako externí programátor a správce počítačové sítě. Celé zadání pro implementaci datového můstku, který je předmětem této připravil právě programátor e-shopu. Spousty zákonitostí e-shopu budou pevně dány, protože síť majitele e-shopu takto funguje a některé věci jsou neměnné. Často se v takových případech používá fráze „protože to jinak nešlo“, která se stala jistým klišé většiny IT pracovníků a toto zná i autor této práce.

Tento můstek se nevytváří jen jako demonstrace skutečnosti, že datové můstky lze konstruovat jinak než pomocí databázových procedur, vlastních skriptů nebo jiných programů na míru ale že se při průzkumu www stránek našla technologie, která se jeví velice zajímavě, použitelně pro tyto procesy a je hlavně i cenově z politiky licencování dostupná ale i jako hlavně potřeba vylepšit některé procesy ve firmě majitele e-shopu a tato technologie bude reálně použita a nasazena do produkce.

Tato diplomová práce bude reálně zabývat načtením skladového hospodářství v softwaru Money S3. Datový přenos v tomto případě bude pouze jednosměrný. Cílový systém bude dnes rozšíření open source e-shopový systém Presta Shop.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem je zvolit přenos dat mezi takovými systémy, aby tato práce nepůsobila jen jako demonstrace použití Knime Analytics platform (dále jen Knime), ale aby řešila nějaký konkrétní problém využitelný v praxi. Tím konkrétním problémem je přesun údajů o skladovém hospodářství, vedeným v aplikaci Money S3.

Prvním cílem tedy bude načíst XML feed z aplikace Money S3. Po načtení dat transformovat výstup do použitelné podoby. V dalším kroku kroku po transformaci a úpravě dat v Knime, hrát data do dočasné tabulky. Ta poslouží pro API rozhraní PrestaShopu jako zdroj dat, která se do e-shopu naimportují.

2.2 Metodika (nástřel jak by to mělo fungovat)

1. Klient provede export dat z Money S3.
2. Dle nastavení naplánovaných úloh na serveru (CRON) se spustí KNIME skript, který daný XML feed načte a zpracuje do dočasné tabulky.
3. Dle nastavení naplánovaných úloh na serveru (CRON) se spustí PHP skripty, které zajistí import dat z dočasné tabulky do PrestaShopu přes PS API.

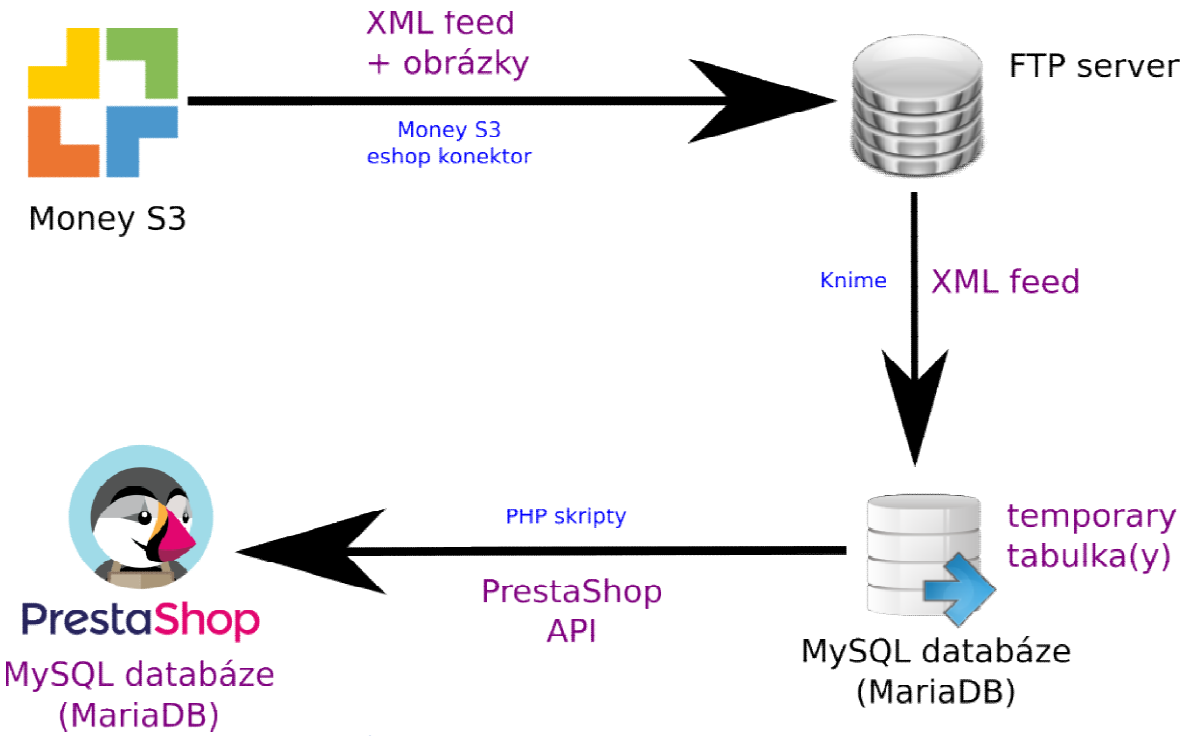


Figure 1 - Schéma implementace můstku

Teoretická východiska

2.3 Představení nástroje Knime Analytics platform

Knime Analytics platform (dále jen Knime) představují autoři aplikace jako platformu pro snadnou analýzu a manipulaci s daty. (1)*"Our KNIME Analytics Platform is the leading open solution for data-driven innovation, designed for discovering the potential hidden in data, mining for fresh insights, or predicting new futures. Organizations can také thein collaboration, productivity and performance to the next level with a robust range of commercial extensions to our open source platform"*. Aplikaci lze tedy použít jako nástroj pro manipulaci s daty jako takovými, hledání nesrovnalostí nebo po provedení analýzi dat připravit další výstupy, které nejsou součástí standardního vybavení softwaru, ze kterého data vzejdou. Hlavní předností, kterou u této aplikace vnímám je grafické rozhraní, které je jednoduché a přehledné pro běžného uživatele.

Zásahní výhodou je možnost spuštění nástroje pro uživatele jiných operačních systémů, než je Windows. Knime je dostupný pro uživatele operačních systémů Linux nebo Mac OS.

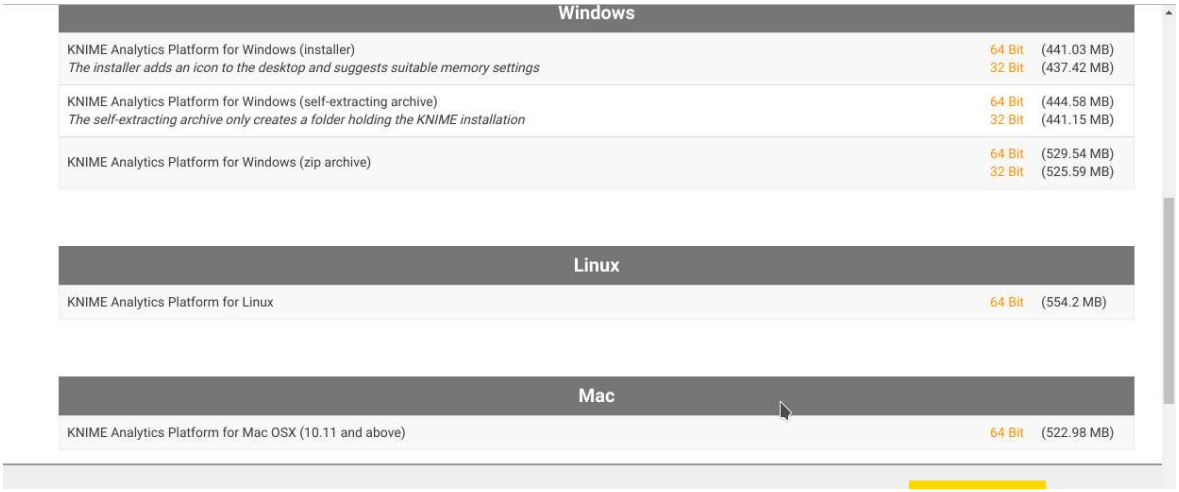


Figure 2 - stránka download

2.3.1 Licencování

Knime je licencován pod licencí GNU General Public License, Version 3. To znamená, že takový software je možné užívat bezplatně a to i v komerčním prostředí. Jenom jako zmínku, Knime dále nabízí nástroje jako Knime Server. Tento nástroj je vhodný pro širší využití Knime Analytics platform pro více uživatelů a řešení správy jednotlivých Knime schémat ukládaných na vzdáleném uložitšti.

2.3.2 HW nároky

Přímo na stránce HW nároky na Knime dohledatelné nejsou, naštěstí se jeden z přispěvatelů na tyto nároky ptal a bylo mu odpovězeno administrátorem, cituji:

(2) *„KNIME requires a reasonably modern PC running Linux, Windows (XP and later), or Max OSX. Sophisticated graphics hardware is not needed, multi core systems a plus as KNIME makes use of multiple cores. The available hard drive space (NOT main memory) limits the amount of processable data - several tens GB free space are*

recommended. Main memory should be 1GB or above, on 32bit systems up to 1.5GB can be used, more on 64bits systems.“

V dnešní době tyto nároky odpovídají normálnímu kancelářskému počítači. Při zkoušení nástroje se opravdu projevilo, že paměťové nároky nejsou hrozně veliké, více záleží na místě na disku. Knime jak jsem zjistil spoustu věcí cachuje (více v kapitole metodiky práce) hlavně při práci s velkým objemem dat a pokud by toto mělo jít výhradně přes paměť, museli by uživatelé Knime mít desítky GB paměti. Pro urychlení procesování dat určitě stojí za zvážení mít počítač vybavený SSD diskem kvůli vyšší rychlosti zápisu a čtení oproti klasickým plotnovým diskům. Na testovacím osobním počítači vybavený plotnovým diskem, zároveň i druhým notebooku s ssd diskem jsem nezaznamenal až tak velkou prodlevu (víc v testování aplikace – prodlevy mezi zpracováním dat).

2.3.3 Interface aplikace

Při spuštění aplikace prve uživatel vybírá pracovní prostor. Tím je složka, kam si Knime ukládá všechna své projekty a lokální pracovní soubory. Po výběru pracovního adresáře se spustí hlavní rozhraní Knime.

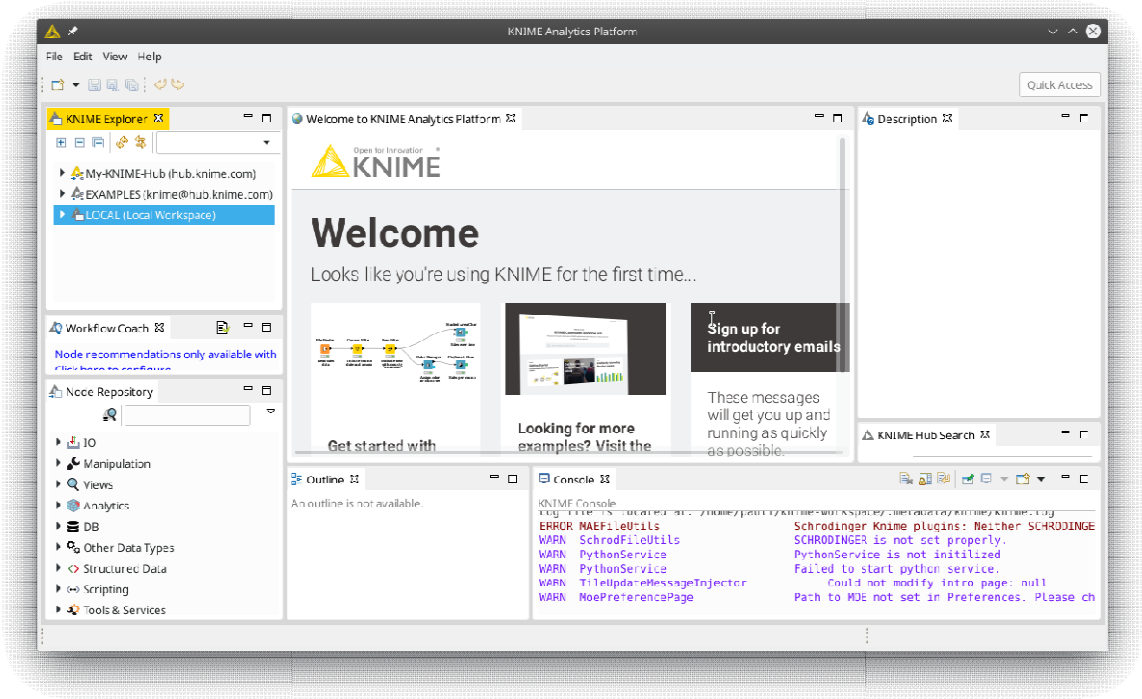


Figure 3 - Knime interface

Inteface se skládá ze začátku z několika okýnek. První vlevo nahoře je KNIME Explorer.

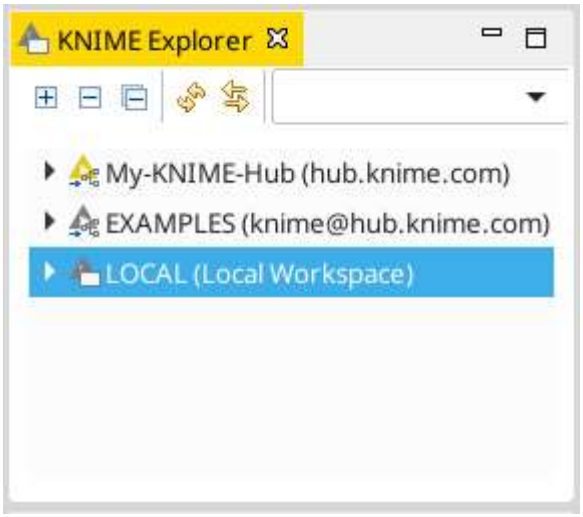


Figure 4 - Knime Explorer

Knime Explorer obsahuje seznam projektů. Jsou zde projekty uložené vzdáleně na Knime serveru nebo projekty uložené lokálně. Po prvním spuštění je zde připojen server s příklady. Ty demonstrují možnosti použití Knime. Pro začátek bylo zjištěno, že pro snadnou orientaci v Knime pluginech jsou tyto příklady zpracované v dostatečné kvalitě. Už jen hlavně proto, aby se uživatelé naučili správně hledat jednotlivé nody (nástroje ve schematu).

Při přihlášení do Knime hubu lze využívat vlastní osobní prostor na vzdáleném uložišti. Při rozkliknutí Knime Hub vyskočí na uživatele okýnko, které ho vyzve k přihlášení skrze prohlížeč. Okno prohlížeče se otevře automaticky. Stejně tak může vytvořit úplně stejně svůj účet.

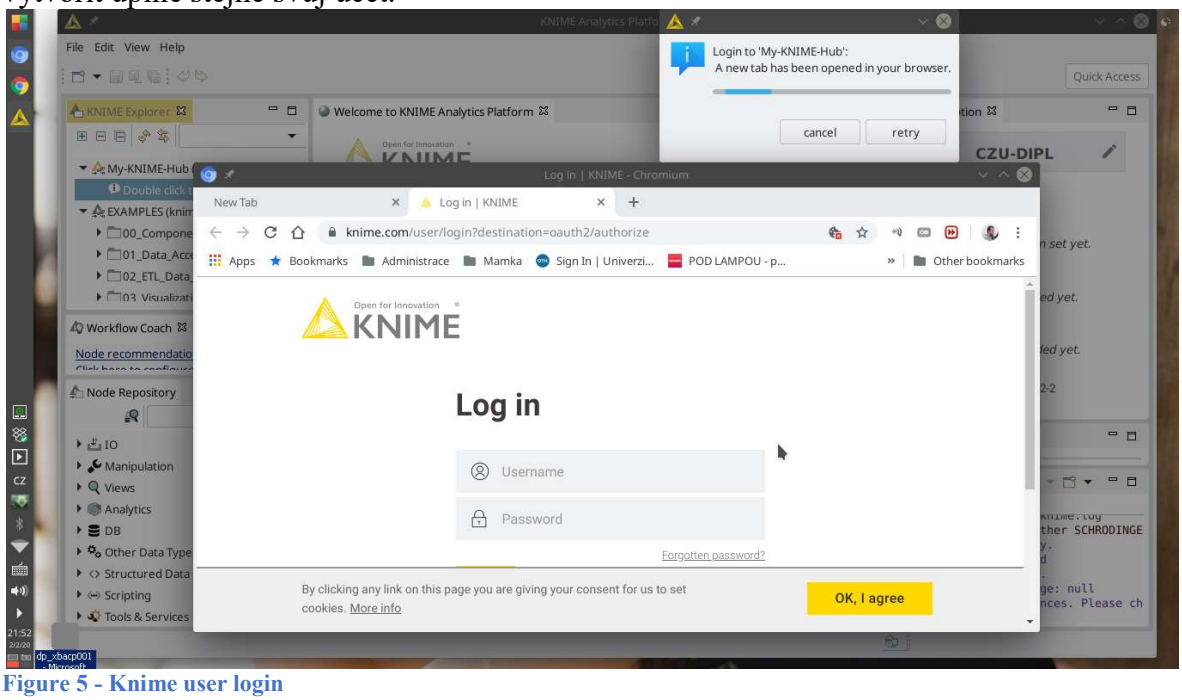


Figure 5 - Knime user login

Vytvoření uživatelského účtu je jednoduché a stejně jako samotný nástroj, zdarma. Po zadání svých osobních údajů jako jméno, příjmení a e-mailový účet na e-mail je odeslán potvrzovací e-mail. Při použití potvrzovacího linku z e-mailu si uživatel může vytvořit svoje heslo. Stále se však musí přihlásit.

Po přihlášení se Knime Explorer změní. Zobrazí se zde dva adresáře. Public a Private. Zde může stejně jako v lokálním úložišti ukládat své projekty.

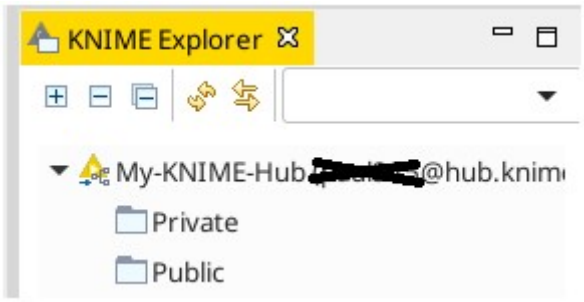
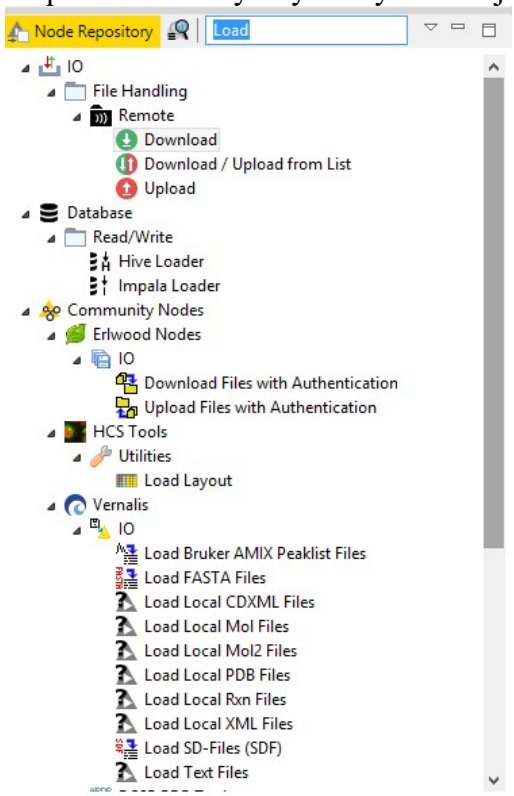


Figure 6 - Knime explorer after login

2.3.4 Načítání dat

V prvotní fázi bylo vyzkoušeno, jakou terminologii Knime používá k pojmenování nodů. Proto byla s ohledem na potřebu načíst zdrojový soubor zvolena následující klíčová slova: Load, Read, Input.

Při použití Load byli výsledky následující:



Obrázek 1 - repozitář nástrojů (Node)

Už jen při kliknutí na jakýkoliv Node, Knime v pravé horní části okna zobrazuje popis a možnosti použití konkrétního Node. Z nabízených Node byl vybrán "Load Local XML Files". Podle popisu načítá jednotlivé XML soubory z místního úložiště na disku. S ohledem na to, že XML soubor z Money S3 nejdřív je stažen do počítače je taková varianta přijatelná.

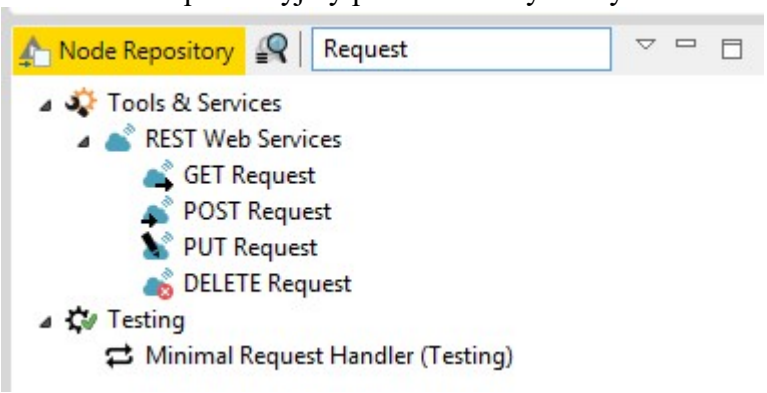
Další testované klíčové slovo bylo Read. Výsledky hledání byli o dost rozsáhlejší než v předchozím případě. Byli zde i nabízeny možnosti jako přímého načítání dat z většiny známých typů databázových systémů. Mezi výsledky se i vyskytl další použitelný Node, pod označením "XML Reader". Podle popisu nabízí sice načítání jen jednoho souboru naráz, ale dokáže ten soubor načítat i z URL adresy. Narozdíl od předchozího Nodu "Load Local XML Files" se tento Node jeví užitečněji. Nabízí se tak možnost řešení, kdy by se XML feed nahrál na webový server, ze kterého by si Knime XML Feed jen stáhnul a rovnou mohl používat aktuální data.

Při testování slova Input z výsledků se jevil zezačátku použitelný Node "Input File". Při otestování Node jeho výsledkem bylo jen poskytnutí cesty k adresáři. S ohledem na předchozí testované nody, které umí už pracovat dál se strukturou XML se tato cesta sice nejeví jako slepá ulička, ale přibylo by několik kroků navíc, než by se Knime dostal ke skutečnému zpracování souboru.

2.3.5 Podporované formáty

Hned na úvod je přiložena následující citace ze stránek autora. (1) *"The IO category contains parser nodes that can parse texts from various formats, such as DML, SDML, PubMed (XML format), PDF, Word, and flat files. Parsing and reading the data into KNIME is the first step which has to be accomplished. The output of all parser nodes is a data table consisting of one column with* DocumentCells. *Each* DocumentCell *contains one document. This list of documents can then be used as input by all nodes of the enrichment category. The DML and SDML format are XML based formats to represents texts in a structured way. Texts available in other XML based formats can be transformed into SDML easily by the usage of XML nodes, provided by the KNIME XML plugin."*

Konkrétní odstavec popisuje pouze načítání dat pomocí vstupu ze souboru. Při průzkumu dalších nodů, viz. odstavec 1.3.5 bylo dokázáno, že Knime umí kromě souborových vstupů připojovat se rovněž na databázové servery. Při odhalení této možnosti připojení byla naráz vyzkoušena další klíčová slova, jako Request, POST, GET, PUT nebo REST na otestování, jestli Knime umí přímo přistupovat na API rozhraní. Už při zadání slovíčka "Request" vyjely povzbudivé výsledky.



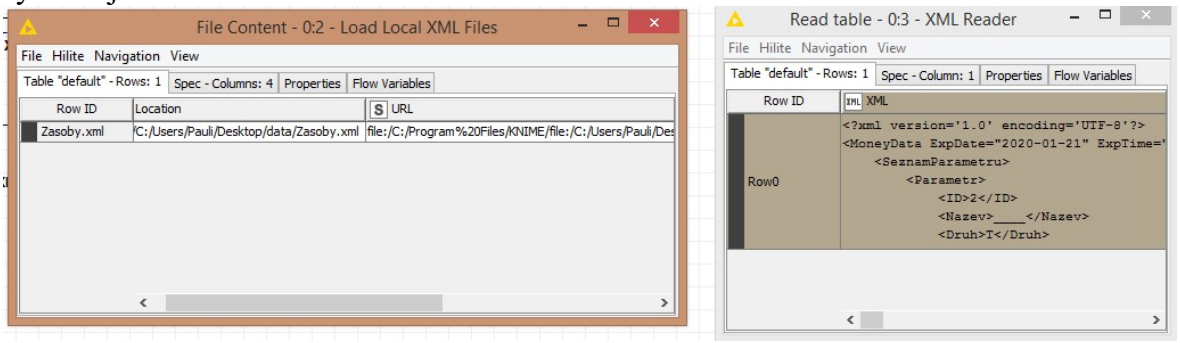
Obrázek 2 - vyhledávání request

Při průzkumu nastavení jednotlivých Node bylo odhaleno, že Knime dokáže přímo poslat jednotlivé typy requestu na webové služby a přímo stahovat výsledky, případně zpracovaná data rovnou pomocí API nahrát. Toto odhalení posouvá možnost řešení importu značně kupředu. Je však nutné mít na paměti, že webové služby jsou schopné přijímat větší objem dat opakovaným voláním a to by mohlo problém. Webové servery,

které přijmou větší objem požadavků mohou dočasně další požadavky blokovat z důvodného podezření, že by mohlo jít o DoS útok. Tyto náležitosti jsou však řešitelné právě využitím PHP skriptů, které by využívali rozhraní přímo na serveru a pro takové případy se dá nastavit bezpečnostní vyjímka. Víc o tom až v samotné implementaci.

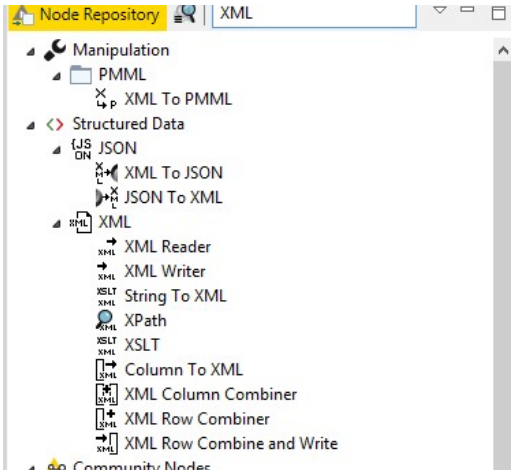
2.3.6 Možnosti zpracování dat

Pro využití Knime je zásadní kromě schopnosti data načítat i s daty manipulovat. V první fázi, když Knime načte soubor, načte si ho jako jeden velmi dlouhý řádek s daty. Od aplikace s grafickým rozhraním je očekáváno z prvu mnohem více. I v tomhle momentě přišlo zásadní srovnání, jak se chová výše popisovaný "XML Reader" a "Load Local XML Files". XML Reader načetl už samotný XML soubor. V případě Load Local XML Files nodu načetl jen cestu k souboru ale nikoliv jeho obsah. S ohledem na potřebu načíst jediný soubor v tomto konkrétním případě se jeví použití dále už jen XML Readeru jako výhodnější.



Obrázek 3 - Načítání XML pomocí Load Local XML FIles a XML Reader

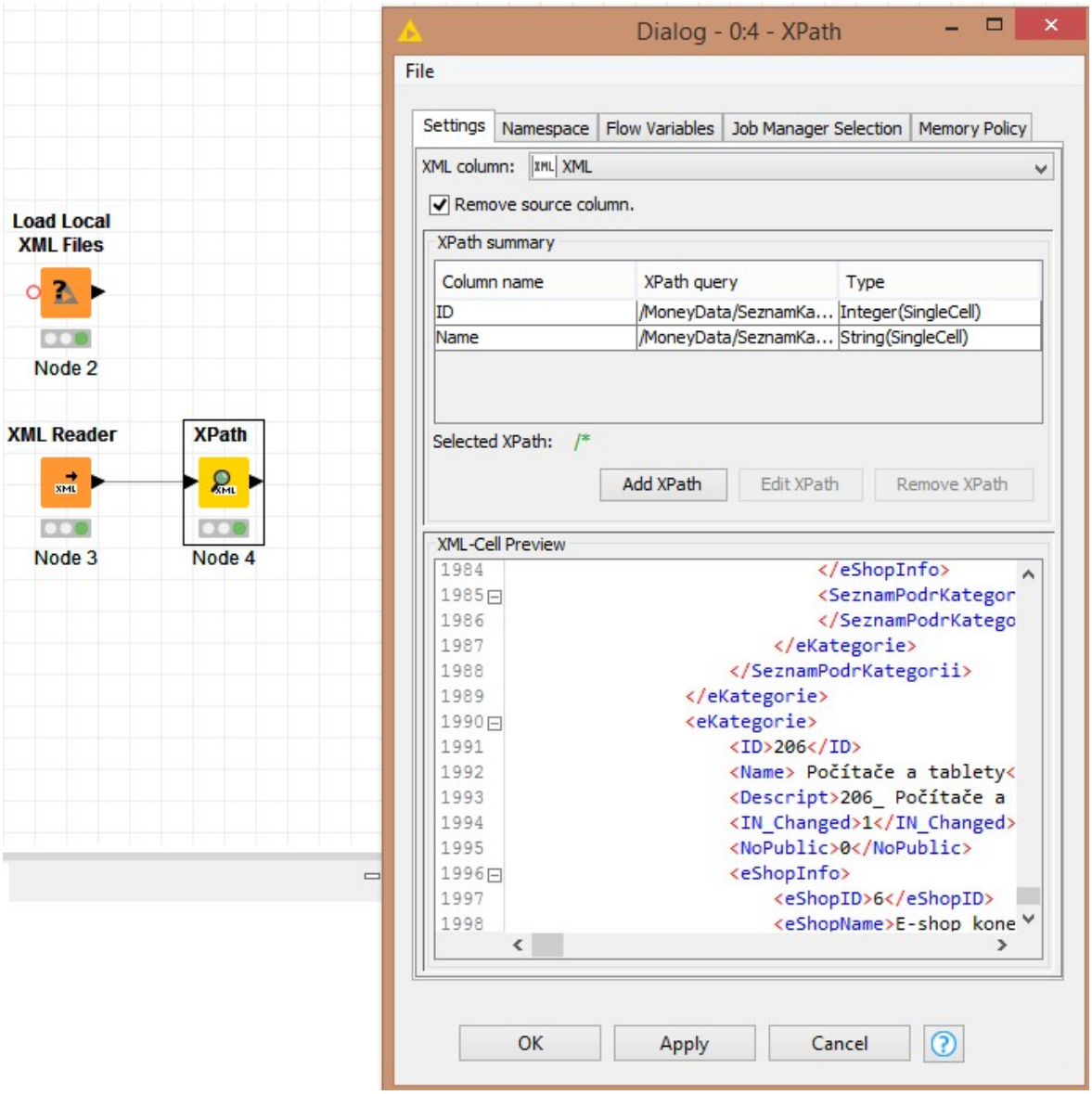
Aby s daty šlo lépe manipulovat, je potřeba prozkoumat jaké další Node v Knime použít, aby byla vidět lépe struktura dat. Konec konců současná podoba v XML Reader připomíná zobrazení XML souboru v pokročilejším textovém editoru. Stejně jako při prozkoumání nástrojů, manipulující s XML soubory při vyhledání klíčového slova XML a zároveň pokusu hledání komunitního řešení na Knime foru. Použitelných nástrojů z prvu bylo nalezeno docela hodně.



Obrázek 4 - Podpora práce s XML

Zajímavé jsou i automatické možnosti převodů XML na JSON, pokud by cílový systém tento formát požadoval. Podle dokumentace je doporučeno, pokud není již zvoleno

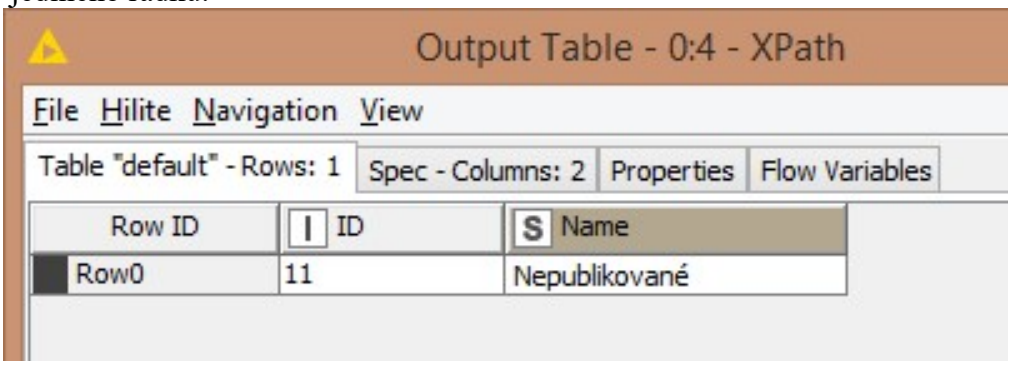
v XML Reader Node, použít jako další prvek schématu Xpath a definovat, co a jak má Knime dále načítat.



Obrázek 5 - Demonstrace XPath

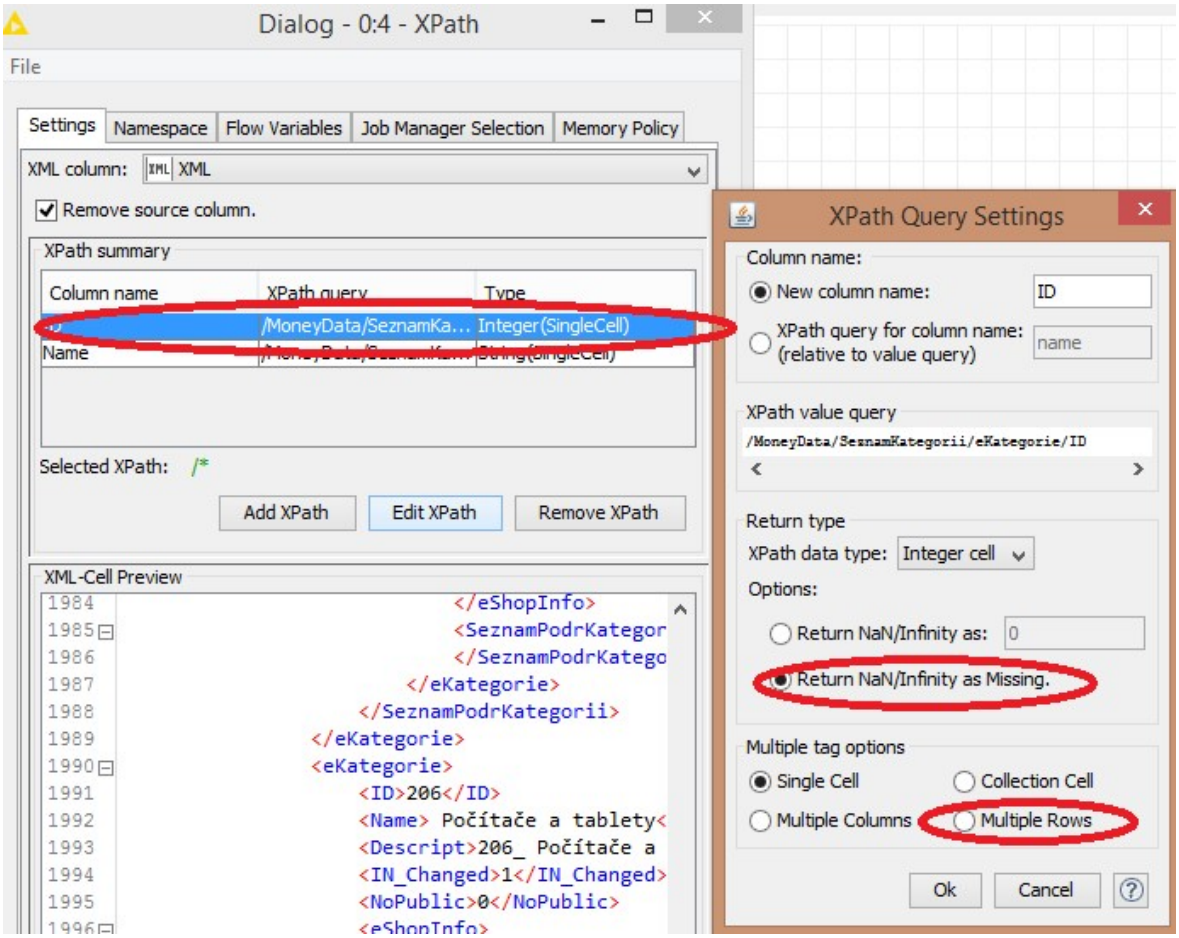
V Xpath summary musí uživatel nastavit, jaká data chce z XML souboru načítat a jaká je cesta ke konkrétním údajům. Knime v tomhle napomáhá značně už zobrazením náhledu. V náhledu může uživatel kliknout na konkrétní údaj, zvolit "Add Xpath" a Knime již v dalším dialogu předvyplní, jak má konkrétní cesta k datům vypadat. Stejně tak uživatel by neměl zapomínat na výběr "typu" hodnoty. Typ udává takzvaný datový typ. Datový typ určuje, jestli konkrétní údaj obsahuje text (řetězec), celé číslo, desetinné číslo. Datových typů Knime dokáže zpracovávat celou řadu, dokonce i binární data, při načtení před použitím XML souboru označil sloupec s XML daty jako "XML" nebo i JSON. Více informací ohledně datových typů je popsáno na této stránce - https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-typovy-system-podruhe-datove-typy-string. Stránka se zabývá programováním pro C#, ale pro pochopení datových typů je

tento text dostačující. Bez dalšího nastavení Node Xpath, Knime při načtení soubor rozdělí jen do jediného řádku.



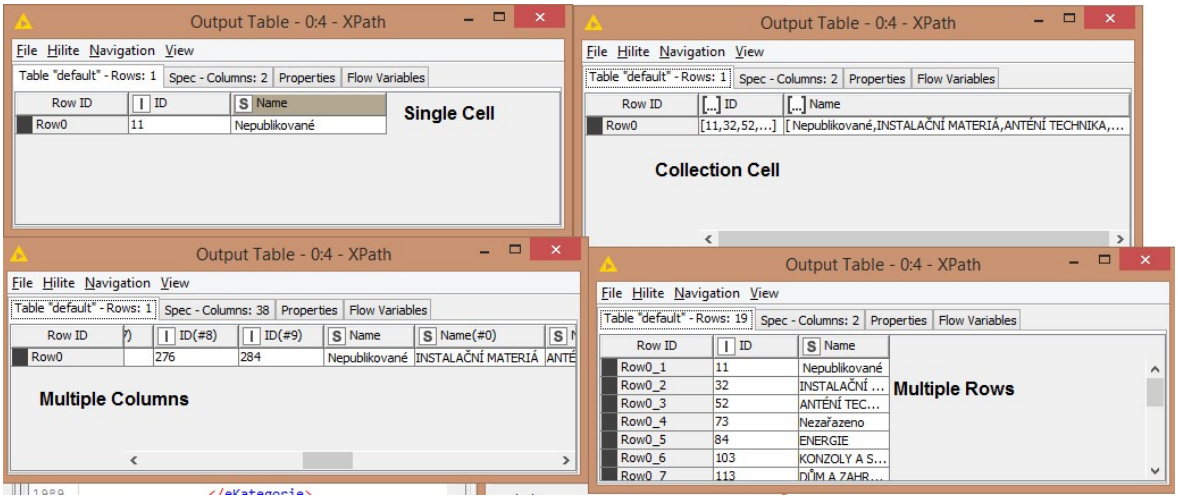
Obrázek 6 - tabulka přes výchozí nastavení

Důvodem toho je pouze skutečnost, že Xpath nemá nastaveno načítání většího množství výskytů konkrétního údaje a tak načte jen první výskyt. Načítání opakovaného použití té samé cesty k datům je pro XPath samozřejmostí dokud záznam bude mít co načítat. Jen je potřeba v nastavení XPath zvolit pár dalších parametrů. V nastavení Xpath uživatel klikne na konkrétní sloupec, klikne pak na tlačítko "XPath". V dalším dialogovém okně, které bylo možné vidět při vytváření jednotlivých "cest" je potřeba zakliknout další volby. První je, aby pro každý řádek záznamů doplnil případně "null", pokud v některém řádku bude hodnota chybět. To by mohlo při dalším zpracování způsobit potíže. Knime by špatně sloučil údaje do řádků v chybějících sloupcích a data by už k sobě správně nepatřila. Dalším nastavením musí uživatel zvolit, jak má XPath zpracovat opakovaný výskyt jednotlivých elementů v XML. Knime výstup načítá do tabulky a podle toho lze vybrat možnosti, jak má tuto tabulku pro další zpracování vygenerovat.



Obrázek 7 - Nastavení XPath Summary

Ve výchozím nastavení je vidět, že Knime načte jen první výskyt, tedy je zvolena volba "Single cell". Volba Collection Cell podle dokumentace vytvoří ve sloupci pole hodnot. Další variantou je pro každý řádek vyjet opakovaný výskyt elementů v XML nové sloupce. Poslední možností je nechat XPath jednotlivé opakované elementy vypsat jako další nový řádek tabulky. V přiloženém obrázku je srovnání jednotlivých výstupů do tabulek. Nevýhodou tohoto nastavování je to, že uživatel musí nastavit formát výstupu pro všechny definované cesty v XPath summary.



Obrázek 8 - Srovnání jednotlivých nastavení

Hlavní je připravit exportovaná data tak, aby je byl cílový systém schopen přijat a zpracovat na své straně. Nejčastěji se v takovém případě provádí nad daty operace jako odstranění nepotřebných dat, úprava tak, aby je cílový systém byl schopen přijmout. V takovém případě jde zase o operace jako přejmenování označených sloupců, odmazání nepotřebných údajů s cílem snížit datový tok.

2.3.7 Výstupní formáty

Knime nabízí mnoho nástrojů pro připojení k datovým zdrojům jak již bylo popsáno výše. Dalším kritériem pro implementaci Knime jako datový můstek je i schopnost udělat z použitých dat i výstup v použitelné podobě pro cílový systém. Pro vyhledávání výstupních nástrojů byla použita kličová slova Write a Output. Už při zadání klíčového slova "Write" Knime nabídl široké spektrum nástrojů, které zapisují data do různých formátů. Nabízí se formáty, PDF, CSV, Excel, Database Writer (MySQL, Oracle, PostgreSQL a mnoho dalších), dokonce HTML, XML nebo i Google sheet.

3 Zdrojový systém

Zdrojový systém je Money S3, který využívá malá firma situovaná v Ústí nad Labem k vedení účetnictví a skladového hospodářství. Komplikací, kvůli které je potřeba vyrobit datový můstek je přenos informací o stavu skladu do e-shopu. Firma nemůže standardně nabízet zboží, které na skladě není. Takový problém by byl v rozporu s obchodními podmínkami o dodání zboží (protože by prodejce nebyl schopen dodat zboží včas, které není naskladněno).

3.1 Předběžná analýza

3.1.1 Podoba exportních dat

Popis výstupu XML feedu z e-shop konektoru bude popsán 2 způsoby. Jednou celý feed jako celek a následně se zaměřením na samotné zboží.

Struktura XML feedu je odlišena pro vyšší přehlednost barevně.

- **Hlavička XML feedu**
- **Označení dat z Money S3**
- **Seznam parametrů Money S3**
- **Seznam kategorií v Money S3**
- **Datová část (zásoby)**

Ukázka feedu XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<MoneyData ICAgendy="40247252" KodAgendy="012" HospRokOd="2020-01-01"
HospRokDo="2020-12-31" description="Export zásob pro XML el.obch.dům"
ExpZkratka="_eShopEZas" ExpDate="2020-01-21" ExpTime="21:45:00"
VyberZaznamu="0" GUID="{CE34BF22-8B20-4596-BD95-CA3CA00BD6C9}">
        <SeznamParametru>
                <Parametr>
                        <ID>3</ID>
                        <Nazev>--1</Nazev>
                        <Druh>T</Druh>
                        <Typ>Vyrobce</Typ>
                        <UzivCode>0</UzivCode>
                </Parametr>
      ..........................
        </SeznamParametru>
        <SeznamKategorii>
                <eKategorie>
                        <ID>11</ID>
                        <Name> Nepublikované</Name>
                        <IN_Changed>0</IN_Changed>
                        <NoPublic>1</NoPublic>
                        <eShopInfo>
                                <eShopID>6</eShopID>
                                <eShopName>E-shop konektor</eShopName>
                        </eShopInfo>
                        <SeznamPodrKategorii/>
                </eKategorie>
      ..........................
    </SeznamKategorii>
        <SeznamZasoba>
      ..........................
        </SeznamZasoba>
</MoneyData>
```

Datová část (XML element "SeznamZasoba") je rozdělena do 6 částí:
Konfigurace produktu (typ zásoba, sazba DPH …)
Stav zásoby
Info - cena, datum nákupu ...
Popis produktu
Kategorie, kam je produkt v Money zařazen
Tayp ceny (s DPH / bez DPH) …..

```
<SeznamZasoba>
        <Zasoba>
            <konfigurace>
                <Druh_zas>Z</Druh_zas>
                <SDPH_Nakup>21</SDPH_Nakup>
                <SDPH_Prod>21</SDPH_Prod>
                <Ev_Min>0</Ev_Min>
                <Ev_Max>0</Ev_Max>
                <Minimum>0</Minimum>
                <Maximum>0</Maximum>
                <UcPrJNakup>V105</UcPrJNakup>
                <UcPrJProde>P503</UcPrJProde>
            </konfigurace>
            <StavZasoby>
                <Zasoba>4</Zasoba>
                <Rezervace>0</Rezervace>
                <Objednano>0</Objednano>
            </StavZasoby>
            <Nak_Cena>503.75</Nak_Cena>
            <Posl_N_Cen>515</Posl_N_Cen>
            <Posl_Nak>2020-01-13</Posl_Nak>
            <Posl_Prod>2020-01-14</Posl_Prod>
            <SklUcet>000000</SklUcet>
            <WWWPopis2>Terestriální přijímač GoSAT GS200DVBT2
Terestriální přijímač GoSAT GS200DVBT2 podporuje nejen starší formáty MPEG2 a
MPEG4, ale taktéž nový H265 HEVC jež se používá pro vysílání v ČR. Přijímač byl
otestován Českými rádiokomunikacemi a dostal certifikát DVBT2 ověřeno. Přijímač umí
přehrávat také video a audio soubory uložené na USB harddisku, nebo flash, které lze
připojit k přijímači přes USB rozhraní. Má-li připojené USB zařízení dostatečnou kapacitu,
lze jej použít k záznamu TV pořadů a lze také využít pozastavení živého vysílání (tzv
Timeshift) Záznamy můžete i naplánovat dopředu pomocí integrovaného časovače. Po
připojení přijímače k internetu lze využít některou z předinstalovaných aplikací (např
počasí, RSS). Přístroj má velmi kompaktní rozměry. Specifikace: DVBT/T2 H.265 HEVC
RF vstupní úroveň -78 ~ -20 dB m IF šířka pásma 7 MHz a 8 MHz Modulace QPSK,
16QAM, 64QAM, 256QAM Video Decoder Format MPEG4 AVC/H.264, H.265 Výstupní
formát 480i/480p/576i/576p/720p/1080i/1080p Výstupní konektory HD, SCART USB 2.0
Podporovaná kapacita do 500 GB Konektor LAN pro připojení k internetu Podporovaná
média MP3, WMA, JPEG, MP4, AVI (Xvid), MKV Napájecí napětí AC220-240V 50/60Hz
Spotřeba 6W (MAX12W) Spotřeba v standby &amp;lt; 0.5 W Rozměry 150 x 97 x 35
mm</WWWPopis2>
            <Vyrobce>GoSAT</Vyrobce>
            <Sleva>0</Sleva>
            <Novinka>0</Novinka>
            <Pripravuje>0</Pripravuje>
            <Vyprodej>0</Vyprodej>
            <ZakazProde>0</ZakazProde>
            <VSleva>0</VSleva>
            <VAkce>0</VAkce>
            <VNovinka>0</VNovinka>
            <PrvaStrana>0</PrvaStrana>
            <UcPohyb>
                <UcProdejk>000000</UcProdejk>
```

```
                    <UcPrijemk>000000</UcPrijemk>
                    <UcVydejky>000000</UcVydejky>
                    <UcVyrobky>000000</UcVyrobky>
                    <UcVDodLst>000000</UcVDodLst>
                    <UcPDodLst>000000</UcPDodLst>
                    <UcManko>000000</UcManko>
                    <UcPreb>000000</UcPreb>
                </UcPohyb>
                <UcVyrobkV>000000</UcVyrobkV>
                <MnInv>0</MnInv>
                <eshop>
                    <IN_Export>6</IN_Export>
                    <IN_Changed>1</IN_Changed>
                    <CisKarty>15861</CisKarty>
                    <CisSkladu>1</CisSkladu>
                    <eSkup>
                        <ID>479</ID>
                        <Name>MPEG-4 T2 H.265 HEVC </Name>
                        <Descript>479_MPEG-4 T2 H.265 HEVC </Descript>
                        <Poznamka></Poznamka>
                        <Changed>1</Changed>
                        <Public>1</Public>
                        <Parent>
                            <ID>478</ID>
                            <Name>SET-TOP-BOXY </Name>
                            <Descript>478_SET-TOP-BOXY </Descript>
                            <Poznamka></Poznamka>
                            <Changed>1</Changed>
                            <Parent>
                                <ID>52</ID>
                                <Name>ANTÉNÍ TECHNIKA</Name>
                                <Descript>52_ANTÉNÍ
TECHNIKA</Descript>
                                <Poznamka></Poznamka>
                                <Changed>1</Changed>
                                <Parent>
                                    <ID>10</ID>
                                    <Name>Všechny
zásoby</Name>
                                    <Descript></Descript>
                                    <Poznamka></Poznamka>
                                    <Changed>0</Changed>
                                </Parent>
                            </Parent>
                        </Parent>
                    </eSkup>
                </eshop>
                <KmKarta>
                    <Popis>Terestriální Přijímač GoSAT GS200DVBT2</Popis>
                    <Zkrat>Terestriální Přijíma</Zkrat>
                    <MJ>ks</MJ>
                    <UzivCode>145843</UzivCode>
```

```
                        <GUID>{3FB3F737-3538-4EF2-BCB4-
655C5159898E}</GUID>
                        <Katalog>14580313</Katalog>
                        <BCTyp>N</BCTyp>
                        <TypZarDoby>N</TypZarDoby>
                        <ZarDoba>0</ZarDoba>
                        <EvVyrCis>1</EvVyrCis>
                        <DesMist>0</DesMist>
                        <Obrazek>Y:\moneys3_fotky_eshop\terestrialni-prijimac-
gosat-gs200dvbt2.jpg</Obrazek>
                        <Obrazek2>Y:\moneys3_fotky_eshop\terestrialni-prijimac-
gosat-gs200dvbt2.jpg</Obrazek2>
                        <Jazyk1>CSY</Jazyk1>
                        <Hmotnost>0</Hmotnost>
                        <Objem>0</Objem>
                        <TypKarty>jednoducha</TypKarty>
                </KmKarta>
                <Sklad>
                        <Nazev>Satelity</Nazev>
                        <GUID>{D58B9F9B-2B42-4072-BC35-
D07101D2B2FA}</GUID>
                        <CenikSklad>0</CenikSklad>
                </Sklad>
                <Skupina>
                        <Zkratka>MPEG-4 </Zkratka>
                        <Nazev>MPEG-4 T2 H.265 HEVC </Nazev>
                        <Zkratka0>ANTÉNY A D</Zkratka0>
                        <Zkratka1>SET-TOP-BO</Zkratka1>
                </Skupina>
                <Posl_Dod>
                        <GUID>{50913ACA-9267-46AF-BE89-
F7BC40380570}</GUID>
                        <Mobil>
                                <Cislo>599526182</Cislo>
                        </Mobil>
                        <EMail>shop@atoselektro.cz</EMail>
                        <ICO>18055761</ICO>
                        <DIC>CZ18055761</DIC>
                        <PlatceDPH>1</PlatceDPH>
                        <FyzOsoba>0</FyzOsoba>
                        <Banka>Československá obchodní banka, a.s.</Banka>
                        <Ucet>169332</Ucet>
                        <KodBanky>0300</KodBanky>
                </Posl_Dod>
                <PC>
                        <SDPH>1</SDPH>
                        <Zaok>N</Zaok>
                        <ZpusobZao>M</ZpusobZao>
                        <Mena>
                                <Kod></Kod>
                        </Mena>
                        <VypPrCeny>V</VypPrCeny>
                        <VychA>N</VychA>
```

```
                    <ZpusobZmA>M</ZpusobZmA>
                    <VychB>Z</VychB>
                    <ZpusobZmB>S</ZpusobZmB>
                    <Cena1>
                        <Cena>730</Cena>
                        <Sleva>0</Sleva>
                    </Cena1>
                    <Cena2>
                        <Limit>0</Limit>
                        <Cena>0</Cena>
                        <Sleva>0</Sleva>
                    </Cena2>
                    <Cena3>
                        <Limit>0</Limit>
                        <Cena>0</Cena>
                        <Sleva>0</Sleva>
                    </Cena3>
                    <Cena4>
                        <Limit>0</Limit>
                        <Cena>0</Cena>
                        <Sleva>0</Sleva>
                    </Cena4>
                    <Cena5>
                        <Limit>0</Limit>
                        <Cena>0</Cena>
                        <Sleva>0</Sleva>
                    </Cena5>
                </PC>
            </Zasoba>
    .....
    </SeznamZasoba>
```

Všechna data jsou poskytována v jediném XML souboru. Knime nemusí takový soubor načítat naštěstí hned 2x. Stačí jej načíst jednou a pro zpracování se definuje více XPath nodů, kde každý z nich už dopředu bude poskytovat potřebná data pro jednotlivé větve schématu.

XML Element SeznamZasoba obsahuje informace o jednotlivých položkách zboží evidovaných v Money S3.

Větev kategorie

4 Cílový system

Cílovým systémem je PrestaShop, verze 1.6. PrestaShop je open source e-shop systém, podle komentáře programátora, který konkrétní e-shop nasadil u této firmy, značně rozšířený hlavně na území České republiky. Výhodou systému je početná vývojářská komunita, která pomáhá řešit dopředu mnoho častých problémů, které se objeví právě při nasazování systému.

4.1 Předběžná analýza

Cílový systém je již integrován s dalšími systémy. Prvním z nich je import od dodavatele zboží. Pokud zboží poskytované dodavatelem nemá konkrétní přeprodejce na skladě, může převzít množství na skladě podle dodavatele. Podle komentáře programátora, který poskytuje hlavně data a podklady pro zpracování tohoto datového můstku, není taková náležitost v rozporu se zákonem o účetnictví. Dalším systémem, se kterým je e-shop propojen je export na službu zbozi.cz. Zde se majitel e-shopu snaží navíc zviditelnit rovnou poskytováním informací o produktech, které nabízí na svém e-shopu. Zboží.cz hlavně nabízí pak i cenové srovnání konkrétních produktů.

4.1.1 Průzkum možností importu dat

Účetní systém Money S3 má vlastní eshopový modul "eshop konektor". Ten z Money S3 vygeneruje XML feed a i spříslušnými obrázky nahraje data na FTP server nebo do příslušné složky. Takový XMl feed pak je potřeba nahrát do PrestaShopu.

1. Nahrát data přímo do databáze eShopu
2. Nahrát data přes API PrestaShopu

Druhá varianta je vhodnější a tou se bude implementace můstku zabývat. Varianta první je sice rychlejší, ale je vhodná pouze, je-li jeden klienta, který má svůj eShop. Proč? Data v importu je třeba tzv. "propsat" do tabulek. Jenže seznam tabulek, kde je třeba provést záznam se mění dle toho, jak moduly PrestaShop má nainstalované. Různé moduly = základní tabulky + různé tabulky daných modulů. Při větším počtu klientů by byla další implementace můstku u nich náročná. Proto využijeme API PrestaShopu pro import. Dané API samo zapíše data tam, kde je to potřeba. Takže je tento import bezpečnější, ale druhou stranu časově náročnější. Výhodou však je fakt, že není potřeba řešit jako při zápisu do databáze, kam všude je potřeba data propsat. Tuto náležitost řeší API a není potřeba pak import pro jednotlivé klienty upravovat.

4.1.2 Podoba importních dat

Prvním kritériem, které integrátora takového můstku zajímá je vazba mezi daty zdrojového a cílového systému. Často je nutné i takovou vazbu vyrobit, protože systémy které nejsou navrženy na přímou komunikaci s jiným systémem s tímto nepočítají. V PrestaShopu bude potřeba řešit vazby hned dvě. V prvém případě musí existovat v PrestaShop přesně stejně dané kategorie zboží, jako eviduje Money S3. Pokud struktura kategorií zboží nebude stejná, import nebude pracovat správně nebo se neprovede vůbec.

Druhým kritériem je potřeba zboží správně do kategorií zařazovat. Money S3 tyto údaje sice poskytuje, bohužel v nepoužitelné podobě pro cílový systém. Knime bude potřeba strojově naučit, jak správně hodnotu kategorií upravit, aby import jednotlivých položek přijal. Podle autora e-shopu jediná použitelná vazba mezi kategorie z e-shopu a Money S3 je její název. Často ale bývají podobné vazby slabé. Stačí jediný překlep v názvu kategorie, nebo její přejmenování v Money S3 a e-shop bude při dalším běhu importu dat bude kategorie přepisovat.

Třetí nesnází importu je pak struktura kategorií u samotného zboží. Money S3 strukturu a cestu kategorie zboží poskytuje v obráceném pořadí narozdíl od požadovaného formátu v PrestaShop. Celý problém se jeví jen jako obrácení řetězce, ale tím zde celá práce bohužel nekončí. PrestaShop vyžaduje jednotlivé cesty kategorií oddělené znakem "|".

Další potřebná úprava spočívá v automatickém přepočtu ceny s DPH podle toho, jestli je konkrétní zboží s DPH nebo nikoliv. Do PrestaShopu budou všechny položky importovány nakonec jako bez DPH ale s přepočítanou cenou, pokud je nebo není. Další komplikací je délka názvu zboží. Money S3 má tento vstupní řetězec ve svém systému příliš krátký. Proto firma názvy celé ukládá v Money S3 do poznámky . To znamená že pro kolonku název pro změnu bude muset proběhnout zkrácení textu na požadovanou a možnou přijatelnou délkou samotným PrestaShopem, to činí 255 znaků.

5 Implementace v Knime

Podle analýzy zdrojového systému je patrné, že zdrojem dat bude XML feed z aplikace Money S3, jelikož přes svůj modul e-shop konektor nejspíš jiným použitelným způsobem data nelze získat.

5.1 Načtení dat

V prvé řadě je potřeba vyřešit přímo načtení samotných dat. Data jsou k dispozici pro Knime ve formě XML souboru. Data obsahují informaci o nastavení aplikace Money S3, dále obsahuje informace o kategoriích jednotlivých kusů zboží a nakonec obsahují i údaje o stavu zásoby. Sice je XML element pojmenován jako "StavZasoba", ale kromě informací o stavu zásoby tento element obsahuje kompletní informace o jednotlivých položkách zboží evidovaných v Money S3.

5.1.1 Připojení ke zdrojovému systému

Knime jak se ukázalo, dokáže načítat data ze vzdáleného uložiště (http://, https://), data mohou být uložená na databázovém serveru nebo přímo v počítači ve formě souboru. Data jsou momentálně pouze dostupná ve formě souboru, proto současně bude probíhat zpracování z lokálního uložiště. To znamená pokaždé, než proběhne přenos dat, musí být data poskytnuta aplikací Money S3. Takové řešení je současně v rozporu s možností automatizace přenosu dat pomocí plánované úlohy.

5.2 Zpracování zdrojových dat

Pro samotný můstek v Knime je potřeba si založit nový schéma kde se bude řešit tato náležitost. Zaprvé je vhodné si i uvědomit, které parametry importního můstku mohou být společné a pokud by byly vkládány do více Node opakovaně, při jakékoliv změně je pak údržba a změna například údajů do databáze složitá. Pro projekt lze definovat proměnné. Proměnné mají hned dvě vlastnosti. Zaprvé, lze do ní umístit potřebnou hodnotu. Následně pak lze proměnnou kdekoliv použít, a pokud dojde k její změně, ta změna se propíše po celém schématu. Další použití proměnných je popsáno na stránkách tvůrce aplikace : https://www.knime.com/knime-introductory-course/chapter7/section1/creation-and-usage-of-flow-variables .

Další důležitou vlastností je možnost proměnné lokálně v jiném běžícím prostředí dále přepisovat pomocí volání příkazového řádku. Taková možnost po dokončení Knime schématu a uložení jej na běžící prostředí umožní automatizaci celého importu. To vyžaduje instalaci Knime na cílovém počítači. Dále na tom počítači musí být uloženo samotné schéma, které se má spouštět. Aby jej bylo možné pouštět automaticky, musí být Knime spuštěn pomocí příkazové řádky. V operačním systému Windows lze takové spouštění naplánovat pomocí "Naplánované úlohy". Pro běžící variantu na linuxovém počítači lze takové spouštění nadefinovat přes terminál. Použité parametry v obou případech jsou identické. Parametry příkazového řádku mohou vypadat hned následovně:

```
$KNIME_DIR/knime --launcher.suppressErrors -nosave -reset -nosplash \
        -application org.knime.product.KNIME_BATCH_APPLICATION \
```

```
-workflowFile=$WORKSPACE/ProcessData.zip \
-workflow.variable=rds_host_name,some_db_id.us-east-
1.rds.amazonaws.com,String \
-workflow.variable=rds_port_number,5432,int
```

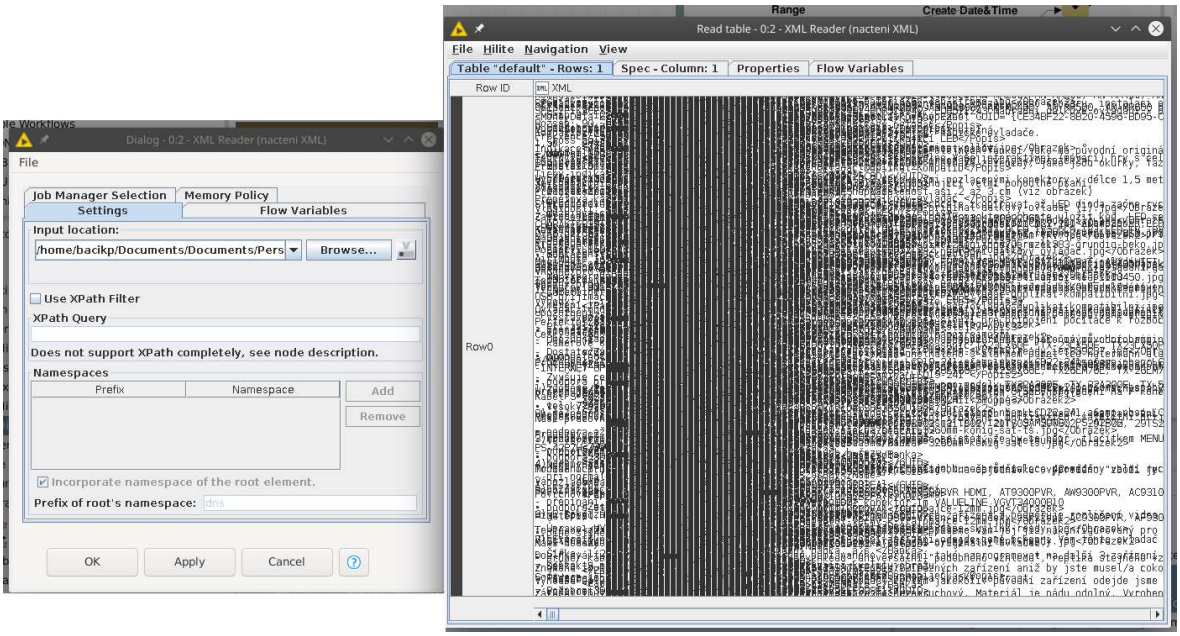
Zde bude následovat popis jednotlivých parametrů:

- -nosave -> schéma po dokončení běhu nebude se ukládat. Schéma v Knime lze při práci uložit nejen jako schéma, ale ukládají se i použitá data, která jsou zde načtena. Tomu je cílem zde konkrétně zabránit
- -reset -> schéma by mělo všechny své Node spustit znova. Pokud ne, tak místa kde jsou obsažená data budou pracovat se zastaralýma datama.
- -nosplash ->Snažíme se aplikaci spustit na pozadí, takže úvodní logo aplikace není potřeba
- -application org.knime.product.KNIME_BATCH_APPLICATION -> tato definice říká, že jde dávkové spuštění aplikace. Tento parametr podle popisu autora je nutné použít.
- -workflowFile=$WORKSPACE/nazevWF -> $WORKSPACE je proměnná, která Knime říká že má hledat soubor ve svém pracovním adresáři podle nastavení. Lze sem vložit i celou cestu k souboru. Dále konkrétní schéma musí být pro běh z terminálu vyexportované pomocí funkce "Export Knime workflow". Ukládá se do formátu "knwf", ale při průzkumu je zjištěno, že to je vlastně jen obyčejný zazipovaný soubor s konkrétním workflow z pracovního adresáře
- -workflow.variable=nazevPromenne,hodnota,typ -> Tady se dá přepsat vnitřní proměnná schématu. Stačí jen název proměnné napsat sem stejně, jako je ve schématu, zapsat novou hodnotu a ještě požaduje použitý datový typ. Možnosti jsou int, string

Konkrétní další informace k používání Knime z terminálu je popsáno pod odkazem https://www.nuwavesolutions.com/batch-execution-of-knime-workflows-part-2/)

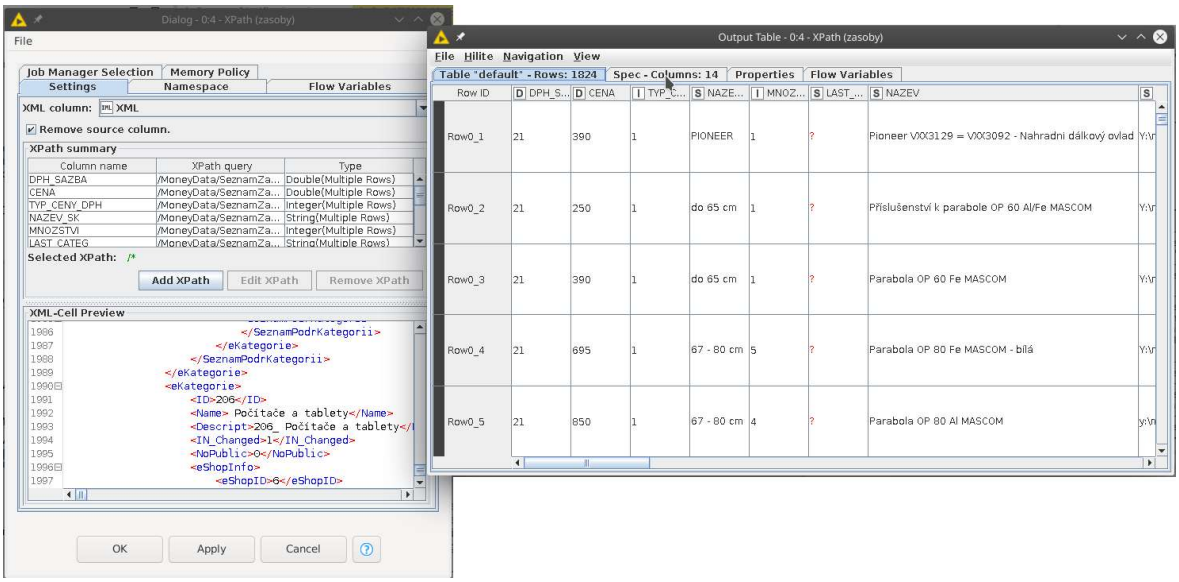
5.2.1 Popis schématu

Celé schéma začne umístěním Node, který má za úkol načíst samotný XML soubor, který Knime má pojmenovaný jako "XML Reader". V nastavení Node existuje i volba nastavit XPath, to znamená, že začne jednotlivé opakující se elementy přenášet do řádků, podle definice určitých pravidel. Tato pravidla mají za úkol říci, jak má soubor XML přečíst a kde hledat jednotlivé řádky položek zboží. Toto nastavení se však lépe připravuje v jiném Node přímo na toto určené. "XML Reader" potřebuje tedy jen nastavit cestu k souboru, který má načítat. Po zadání cesty lze otestovat funkčnost pravým klikem na Node a zvolením volby "Execute". Node mají pod sebou semafor. Červená barva znamená, že Node nelze spustit kvůli chybě nebo nedostačujícímu nastavením. Oranžová znamená, že Node lze zkusit zpustit. Občas se může stát, že nastavení prve není odhaleno jako vadné a projeví se další nečekaná chyba. Tím může být i chybný vstup (například snažíte se nahrát do XML node jiný soubor než XML, nebo XML soubor na vybraném místě neexistuje). Pokud semafor je zelený, Node byl načten do paměti programu a lze s ním pracovat. Nevýhoda většího XML je, že ihned po načtení je v jednom řádku celé XML okem nečitelné.



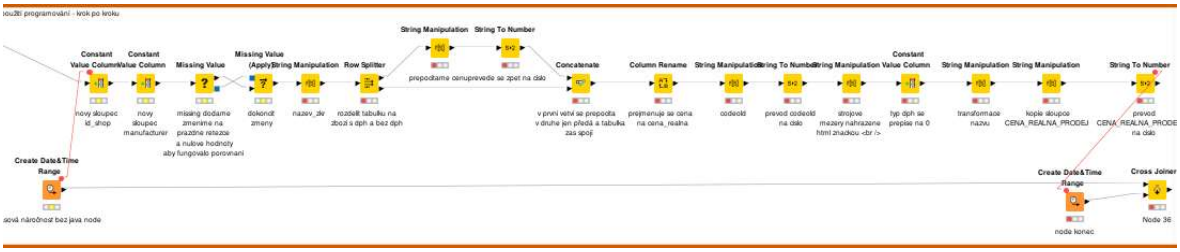
Obrázek 9 - načtení souboru XML Reader

V dalším kroku je potřeba určit, jaké údaje má Knime ze zdrojového soubor načítat, jak vypadají jednotlivé řádky a které údaje chceme načítat. Knime při používání XML Readeru pracuje tím stylem, že požaduje, aby uživatel vybral, které XML elementy má do řádků načítat. Jestli těch řádků může být více nebo má načítat jen jeden výskyt. Jeden výskyt bychom očekávali v případě, že budeme načítat nějaké nastavení ze zdrojového programu. V tomto konkrétní případě jde o načtení samotného zboží pro import do cílového systému. Konkrétní zboží podle struktury v některých případech má ještě navíc další podřízené elementy, viz. popis a ukázka struktury v kapitole "Analýza vstupních dat". Štestí je, že v konkrétním zboží tyto elementy v podelementech mají jen jediný výskyt. Až jedinou vyjímku. Tím je kategorie zboží. Struktura je uložena pod hlavní skupinou v elementu "eSkup", kde už pak názvy jednotlivých kategorií pokračují jako "Parent" a v každém elementu je pak hodnota pod elementem "Name". Něco takového lze řešit programově pomocí rekurzivní funkce, cílem je ale celé toto posunout do Knime. Proto jediný element eSkup vrátíme jako XML a najde se další způsob zpracování. Cílový systém požaduje u zboží poslat skupinu v jednom řádku jako cestu, oddělené znakem |. Další obtíží je fakt, že zdrojový systém má tuto cestu evidovanou od konce. E-shop ji vyžaduje do začátku. V Node Xpath je tedy nutné definovat jednotlivé údaje, které potřebuje cílový systém a všechny vrátit jako typ "Více řádků", jelikož není snahou datového můstku přenést jediný kus zboží. Xpath pak vypíše po nastavení celý XML se zbožím jako tabulku pro další zpracování. Co do dalšího zpracování není potřeba je předávat dál originální XML. To se dá vypnout zaškrtnutím v nastavení "Remove Source Column".



Obrázek 10 - Konfigurace XPath nad daty

Zdrojový systém v XML bohužel přidává i některé chyby. Podle komentáře programátora e-shopu tyto chyby spočívají hlavně v nežádoucích znacích, zbytečných mezerách a pro potřeby webového prohlížeče nutnosti, některé znaky nahradit HTML tagy. Dále e-shop požaduje ceny poslat v jednotné formě. Money S3 tyto údaje posílá tak, jak je má v evidenci. Především jde hlavně o přepočet DPH, pokud je zboží nebo není evidované s DPH. Dále e-shop vyžaduje přidání některých informací navíc, které normálně Money S3 neposkytuje. Většinou jde o údaje identifikující data jako import z Money S3. Knime má na toto několik Node, které umí tyto náležitosti splnit. Jsou to Node jménem "String Manipulation" a "Constant value column". V dalším kroku je tedy snaha přidat do tabulky ke každému záznamu statický sloupec s pevnou hodnotnou. První je ID shopu. Pro potřeby firmy stačí toto zadat pevně danou hodnotou, protože provozuje jediný e-shop a všechno zboží je posíláno jen do tohoto jediného e-shopu. Dalším údajem je krátký řetězec, které přidá do e-shopu informaci, že tato data jsou přijata z Money S3. K tomu programátor e-shopu navrhl kolonu "manufacturer", kde požaduje předat hodnotu "money". Tím e-shop pozná, že jde o data přijaté importem. Protože firma není plátcem DPH, zboží musí být uvedeno bez DPH. V prvním kroku toto zajistí podmíněný přepočet sumy. V druhém kroku pak všechno zboží je napevno označeno jako zboží bez DPH. Dále e-shop požaduje informaci o nákupní ceně. Tuto informaci e-shop vyžaduje, protože je daná navrženým rozhraním ale tato informace je potřeba jen v účetní aplikaci. Proto programátor požadoval, aby se jako nákupní cena předávala cena, za kterou je zboží nabízeno. Jde tedy jen o duplikaci jednoho sloupce v tabulce se zbožím. Celá tato varianta nastavení a požadavků lze i připravit mimo původně navrženého řešení i jednodušeji se znalostí programovacího jazyku Java. Knime umí pracovat interně s programovacím jazykem Java, externě umí využívat i python. Zde tedy bude následovat srovnání kroků, které Knime musí udělat pokud uživatel programovací jazyk nezná proti možnosti řešení pomocí programovacího jazyku Java.



Obrázek 11 - schéma bez programování

Ikdyž byla velká snaha nejít do psaní něčeho co by připomínalo programování, tato snaha selhala. Pro použití string manipulation se uživatel sice lehce nevyhne "používání příkazů", ale toto lze srovnat například s používáním vzorců v MS Excel, kde zadává agregační funkce jako "Sum()". To znamená, že vybere, co se má do Sum načíst a to už pak Sum spočítá. V tomto příkladě je výstupem sloupec, který ale musí být vždy textem. Pak je potřeba převádět zpět takto modifikované číselné sloupce zpět na typ číslo. Zajímavé je, že uvnitř Node "String manipulation" umí používat i matematické operace a s čísly správně nakládat. Výsledek vždy musí jít do řetězce. Zajímavé je i srovnání a doba trvání vytvoření jednotlivých testovaných bloků.

Celý postup spočívá v několika krocích. V prvním kroku se přidávají některé statické sloupce. Jedná se o sloupce id_shop, kde se předává konstantní hodnota 1.

Pak se přidává sloupec manufacturer s hodnotou "money".

Knime neumí správně zpracovávat při porovnání sloupce, které obsahují při zobrazení přehledu červený otazník (?). To znamená, že tyto hodnoty neexistují, chybí. Knime potřebuje aby každá hodnota, která bude nějak porovnávána byla nastavena. Jinak vykazuje chyby při zpracování. Tyto hodnoty jsou v tomto kroku nahrazeny v případě čísel nulou, v případě textu prázdným řetězcem.

Na základě pravidel stanovených programátorem, v dalším kroku se vytváří z názvu zkrácený název na maximální délku 255 znaků.

Pro další manipulaci je potřeba přepočítat zboží s DPH. V tomto kroku spočívá komplikace. Některé položky zboží DPH mají mít, některé ho mít nemají. To určuje sloupec TYP_CENA_DPH, který byl takto ve schématu pojmenován z XML souboru. Lze toto udělat pomoci string manipulation Node ale snaha je zde se vyhnat psaní jakémukoliv kódu, i když to se na konec stejně nepodařilo. Na to existuje node Row Splitter. Ten na základě podmínky je schopen tabulku rozdělit do dvou tabulek, kde v jedné větvi jsou záznamy, které nastavenou podmínku splňují, v druhé tabulce je zbytek, který už podmínce neodpovídá. Použití podmínky lze srovnat s použitím funkce "KDYŽ" v MS Excelu.

Za tímto krokem v tabulce, která podmínku splňuje, probíhá přepočet ceny podle DPH, druhá větev tabulky je nedotčená.

Po dokončení přepočtu je zas potřeba tabulky sloučit. To dělá Node Concatenate. V této fázi se už uživatel nevyhne drobná manipulaci s kódem. Node String Manipulation má naštěstí všechny kusy používaného kódu velmi dobře zdokumentované a všechny příklady použití jednotlivých příkazů jsou ihned dostupné kliknutím na používaný příkaz. Stejně tak uživatel nemusí hádat, jak má zadat vstupní parametr. Jak se píše vstupní parametr zjistí hned tím, že 2x poklepe na konkrétní sloupec, se kterým chce pracovat a Knime ihned vygeneruje kód, jak má zápis vypadat. Stejně tak ve String Manipulation vybírá volbu, jestli se získanou hodnotou vytvoří sloupec nový nebo přepíše nějaký stávající. Jak bylo

zmíněno dříve, vždy musí být výstupem typu text. V případě čísel musí následovat Node String to Number, který převede vybrané sloupce s textem zpět na číslo.

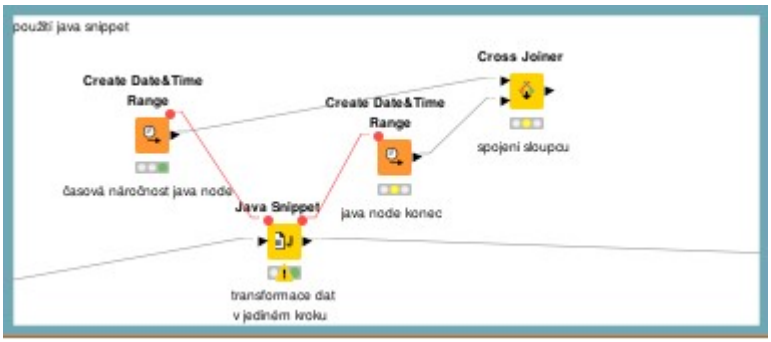
Ve schématu jsou všechny konverze vyrobené naráz hned po sobě. Node String to Number umí převést i naráz větší množství sloupců. Tím se zas sníží počet provedených operací a Knime pracuje pak znatelně rychleji.

Stejně tak "String Manipulation" je použit i k transformaci celého názvu k odstranění znaků na další řádek. Money S3 generuje tyto znaky třemi způsoby. Programově pomocí asci znaku \n, nebo html značky
 nebo ještě upraveného zápisu
. Všechny tyto tři případy musí být odchyceny a přepsány za prázdný text, tedy smazány z názvu.

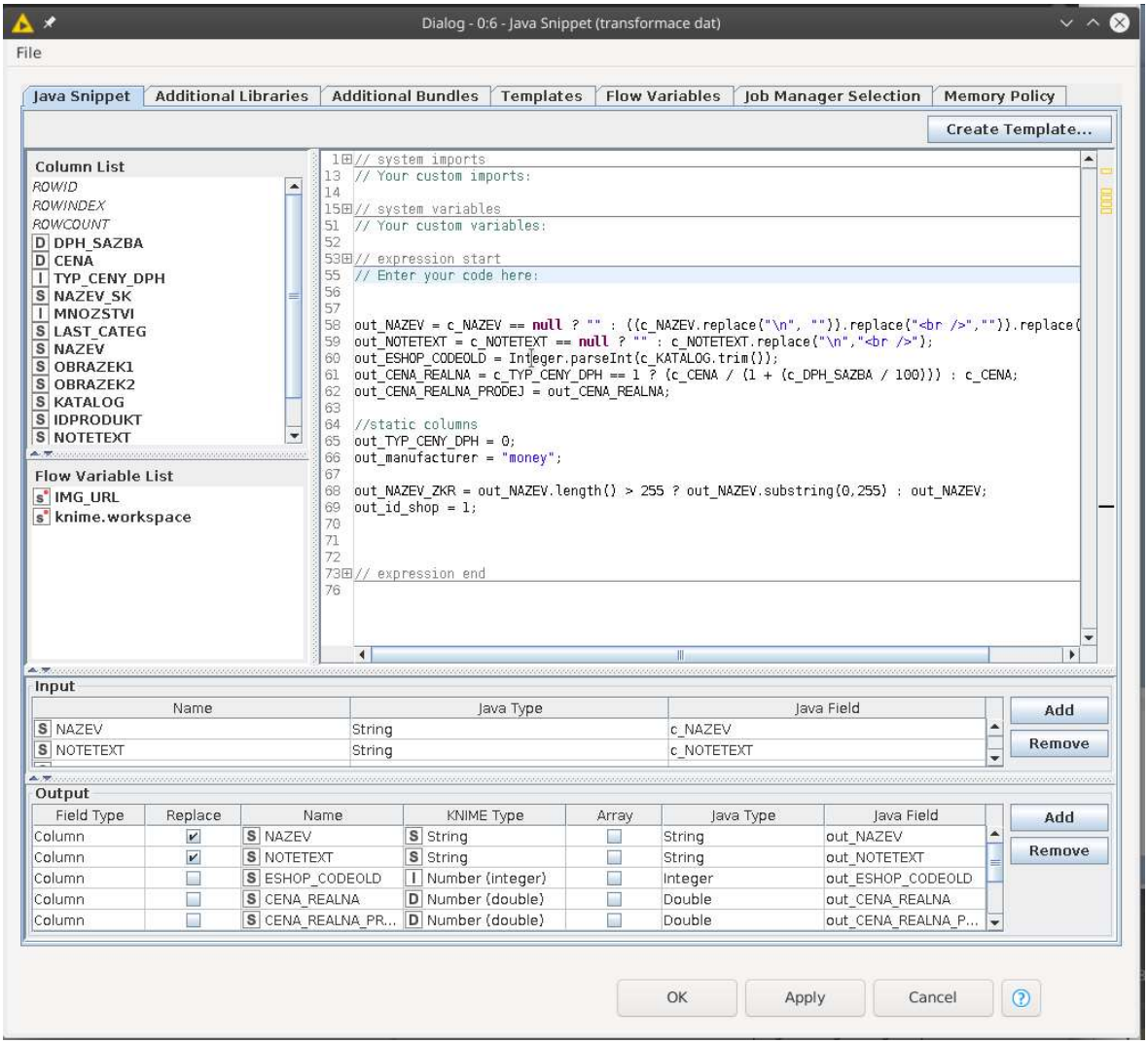
V posledním kroku pomocí string manipulation se vytvoří kopie sloupce CENA_REALNA. Knime nemá v sobě žádný nástroj, který by uměl vytvořit kopii sloupce a toto je jedna z možností, jak tu kopii vyrobit.

Vytvoření bloku bez použití programovacího jazyka s identickým výsledkem trvalo přibližně kolem hodiny času. Se znalostí programování v Java byl návrh rychlejší do 10 minut.

Se základní znalostí programovacího jazyka Java jde toto celé však nahradit jediným Node, který pracuje naprosto identicky, jako řešení na předchozí ilustraci.



Obrázek 12 - použití Java Node



Obrázek 13 - Zdrojový kód v Java Node

Bylo i vyzkoušeno, které řešení je rychlejší. To lze v Knime otestovat například použitím Create Date&Time v momentě, kdy se spouští konkrétní blok. Zabránit spuštění konkrétního data před dokončením sériově je snadné, i když nemá vstupní port. Každý knime Node má vstupní porty pro java proměnné. Každý Node může přenášet proměnné a může je i přijímat. Není však povinnost pokud ty proměnné jsou přeneseny je zpracovávat. Přesto než doběhne předchozí blok, nedokáže se Node spustit, než dokončí práci předchozí Node, který má proměnné poskytnout. Jeden generátor času je umístěn před průběhem testované sekvence Node, druhý časový generátor je umístěn za testovací sekvencí a spustí se po její dokončení. Oba časy pak jen stačí od sebe odečíst. Zdrojový soubor poskytuje celkem 1824 položek zboží. V případě Java Node je měření při tomto množství zboží neprůkazný. Časový rozdíl umí Knime spočítat nejméně v řádu vteřin a Java Node celé zpracování provedl pod jedinou vteřinou. Měření tedy ukazuje "0s". Při měření druhé varianty bez použití programování Knime sice taky nebyl schopen naměřit jedinou vteřinu coby dobu trvání ale bylo vidět, že průchod celého schématu aplikaci už trvalo déle. Je zde i domněnka, že měření použitelné aplikací Knime nedokáže být průchodné v řádu sekund. Konec konců, k tomuto generátory času v Knime nejsou ani navrženy ale pokus to byl zajímavý.

Pro další účely přenosu dat proto nakonec bylo rozhodnuto o použití výsledku z node Java Snippet, který pracuje rychleji se stejným výsledkem.

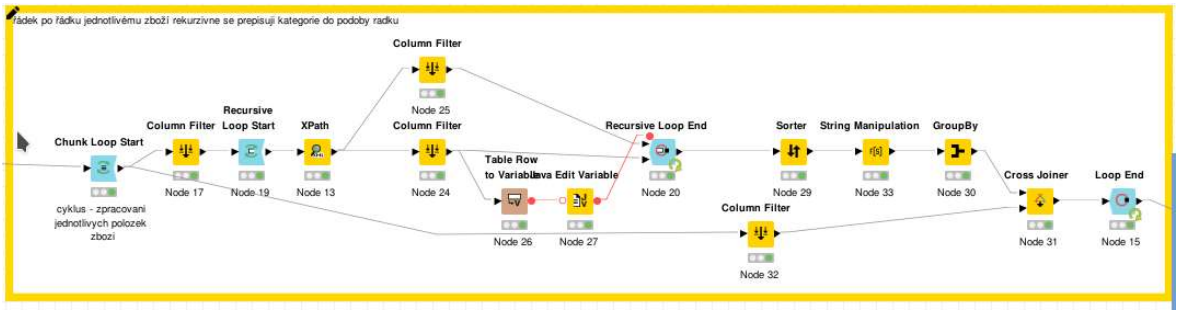
Po dokončení prvních kroků transformací dat je potřeba ještě zpracovat zbývající XML s kategorií produktu. Každá položka zboží má sloupec právě s tímto XML. V prvním kroku je potřeba si rozmyslet, jakým způsobem vytvořit ze struktury XML s kategoriemi, která vypadá následovně:

```
<eSkup>
    <ID>479</ID>
    <Name>MPEG-4 T2 H.265 HEVC </Name>
    <Descript>479_MPEG-4 T2 H.265 HEVC </Descript>
    <Poznamka></Poznamka>
    <Changed>1</Changed>
    <Public>1</Public>
    <Parent>
        <ID>478</ID>
        <Name>SET-TOP-BOXY </Name>
            <Descript>478_SET-TOP-BOXY </Descript>
            <Poznamka></Poznamka>
            <Changed>1</Changed>
            <Parent>
                <ID>52</ID>
                <Name>ANTÉNÍ TECHNIKA</Name>
                <Descript>52_ANTÉNÍ TECHNIKA</Descript>
                <Poznamka></Poznamka>
                <Changed>1</Changed>
                <Parent>
                    <ID>10</ID>
                    <Name>Všechny zásoby</Name>
                    <Descript></Descript>
                    <Poznamka></Poznamka>
                    <Changed>0</Changed>
                </Parent>
            </Parent>
    </Parent>
</eSkup>
```

výsledek, který má vypadat následovně: *Všechny zásoby|ANTÉNÍ TECHNIKA| SET-TOP-BOXY|MPEG-4 T2 H.265 HEVC* .

XML obsahuje několik chyb, které je potřeba rovněž zachytit zavčasu. Těmi jsou hlavně takzvané bílé znaky. To jsou znaky a značky formátování textu, které standardně okem nejsou vidět. Například oddělovače řádků, odstavců pomocí tabulátoru nebo mezery. Z tohoto XML bude potřeba přečíst vše co je v XML elementu "Name" a začít skládat výše zmíněnou větu. Další komplikací je, že text bude obráceně. Jak už je tedy známo, Xpath umí načíst záznamy na stejné urovni do řádků. Tady toto neplatí, další podkategorie je pod-elementem předchozí skupiny, obalené v elementu Parent. Bude tedy potřeba zpracovat každý kus zboží po jednom a načíst jeho strom kategorií. K vytvoření cyklu v Knime, který zajistí že v určité části projde celou tabulku řádek po řádku, použije se Node "Chunk Loop

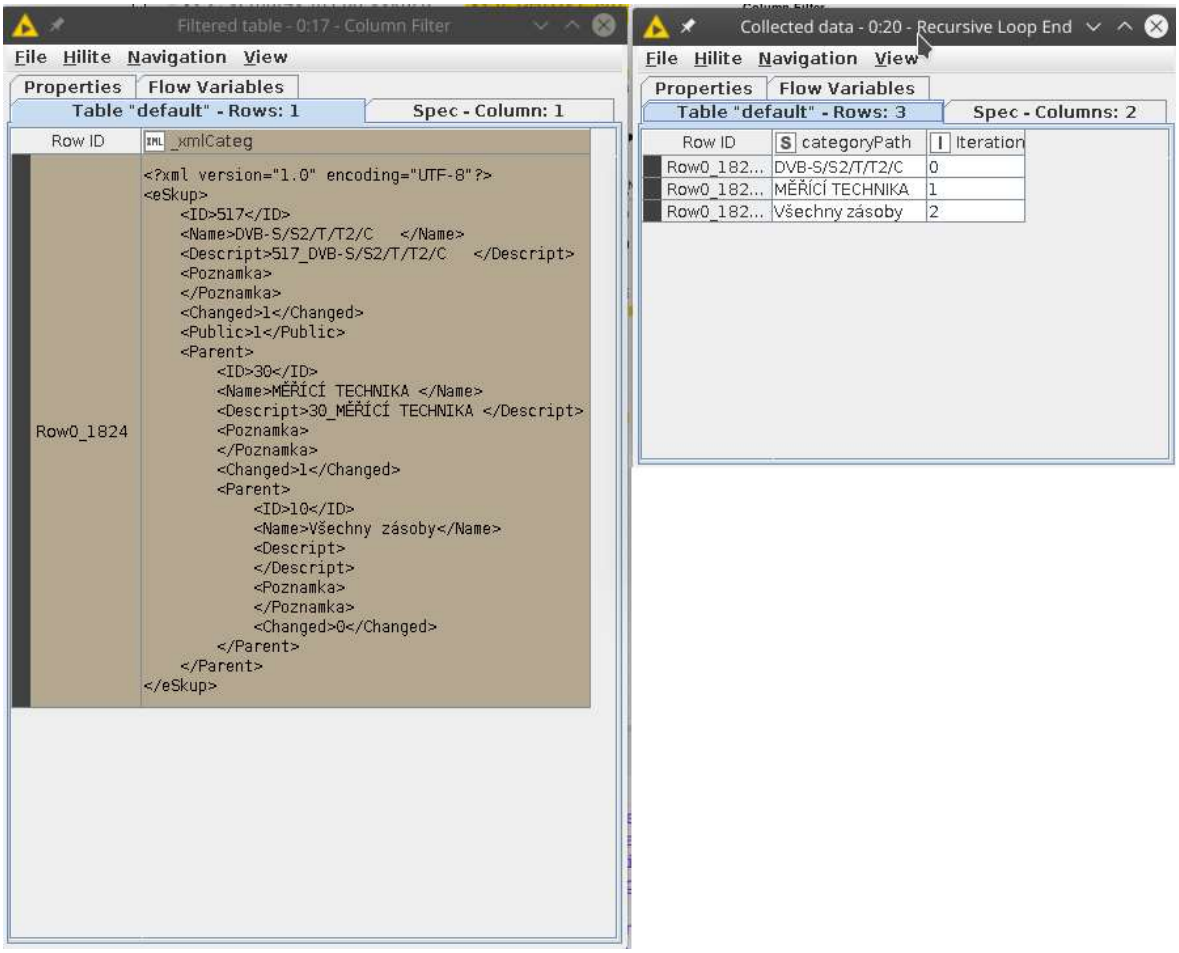
Start". Na konci celého zpracování musí pak být Node "Loop End". Chunk Loop Start právě načítá celou tabulku po řádcích. K načtení kategorie z XML se použije opět Xpath. Ten sice načte první výskyt názvu, ale je potřeba ze "zbývajícího XML" načíst i zbytek cesty ke kategoriím. Taková záležitost se dá řešit pomocí dalšího cyklu. Stejně tak při dalším zpracováním Xpath je potřeba, aby se předal název kategorie z předchozího kroku a zbytek XML bez kategorie, která už byla načtena. K tomu pomůže už druhý cyklus. Cykly samy o sobě nevrací do dalšího průběhu cyklu hodnoty z předchozí iterace. Knime má pro tento účel Node Recursive Loop. To přesně udělá to, co je potřeba. V Node Recursive Loop End jsou 2 vstupy. Jeden sbírá do řádků data z jednotlivých běhů. V druhém vstupu tento Node vyžaduje data, která se mají předat do dalšího běhu. Bez dalšího nastavení v prvním běhu tento cyklus téměř splnil očekávání. Cyklus stále provádí další běhy, ikdyž už nemá žádná další data ke zpracování. Je tedy potřeba ten cyklus zastavit, pokud už nemá co dělat. Podle dokumentace Recursive Loop End v tomto případě lze zastavit předáním proměné, která mu řekne, že už nemá provádět další iterace. Další komplikaci, kterou Knime přímo eliminuje je nutnost zadání maximálního počtu iterací cyklu. Tím se zabrání nekonečnému cyklení, které běžně pro běžící počítač znamená zahlcení paměti a tím pád systému. Výchozí limit 100 iterací bude pro tento případ více než dostatečný. Schéma této části vypadá následovně.



Obrázek 14 - zpracování kategorií

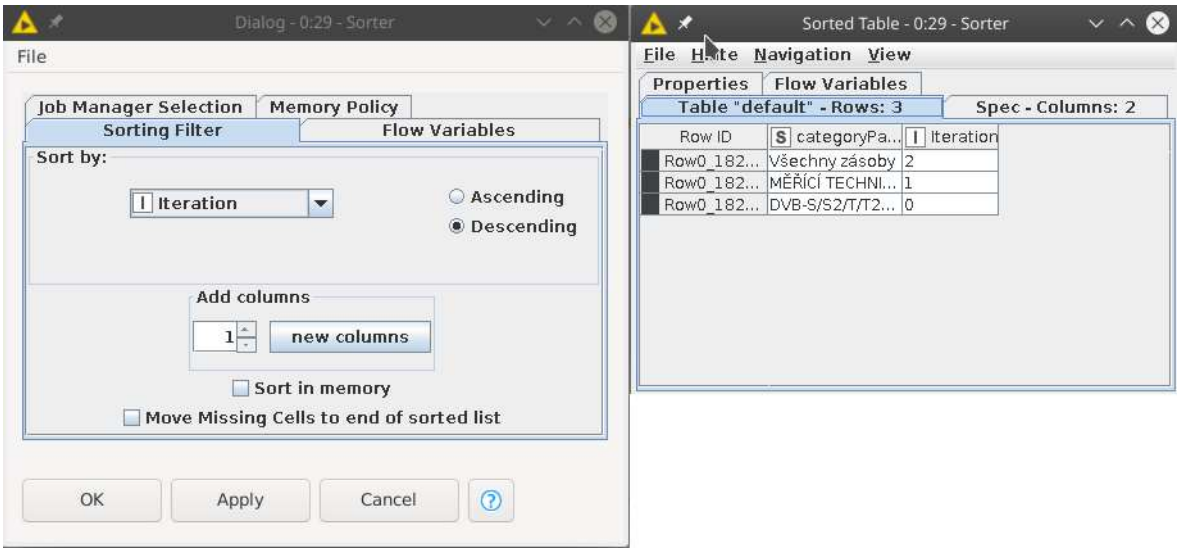
Do rekurzivního cyklu stačí přenášet pouze xml s kategoriemi. Toto je rozděleno, jak lze vidět ve schématu do dvou větví, kde v jedné větvi proběhne převod kategorií na řetězec a v druhém zbytek sloupců se pak jen spojí s výsledkem rekurzivního běhu cyklu. Hned po spuštění rekurzivního cyklu je přidán Node Xpath. Zde se načte první výskyt elementu Name, který bude použitý do řetězce s kategoriemi. Stejně tak se zde načte první výskyt elementu Parent, jehož potomkem je XML s podkategorií. Za Xpath se umístí 2x Column Filter. Jednou se odfiltruje zbývající XML s podkategorií a v dalším případě se odfiltruje načtené jméno. Do Node Recursive Loop End se na první port připojí výsledná tabulka s jedním řádkem, kde je získaný obsah elementu "Name" z XML. Do druhého vstupu se připojí zbývající XML s podkategoriemi. Toto však zcela nestačí. Je potřeba, pokud už v XML nebude nic, což znamená všechny kategorie i podkategorie jsou načteny, zastavit cyklus. Protože Recursive Loop End tento údaj přijímá pomocí Java proměnné, bude potřeba přidat do cyklu další 2 Node. Prvním z nich je převést získaný řádek na Java proměnné. To udělá Node "Table Row to Variable". Podmínkou ale je, že vstupující tabulka musí mít pouze jeden řádek. Jinak Node selže na chybě. Dalším Node je Java Edit Variable. Zde pomocí jednoduchého javovského kódu se otestuje, jestli sloupec obsahující XML něco obsahuje, nebo je prázdný. Pokud je prázdný, musí se do nové proměnné do konce cyklu nastavit hodnota "true". Tomu pak cyklus rozumí jako konec dalších iterací a

skončí s dalším během. Pokud však v XML ještě něco ke zpracování je, může být proměnná na test vracet cokoliv. V tomto případě tam je číslo 1. Když se podíváme na výsledek rekurzivního cyklu, dostaneme tabulku se všemi kategoriemi a dalším sloupcem, který uvádí ve které iteraci cyklu byla konkrétní kategorie načtena.



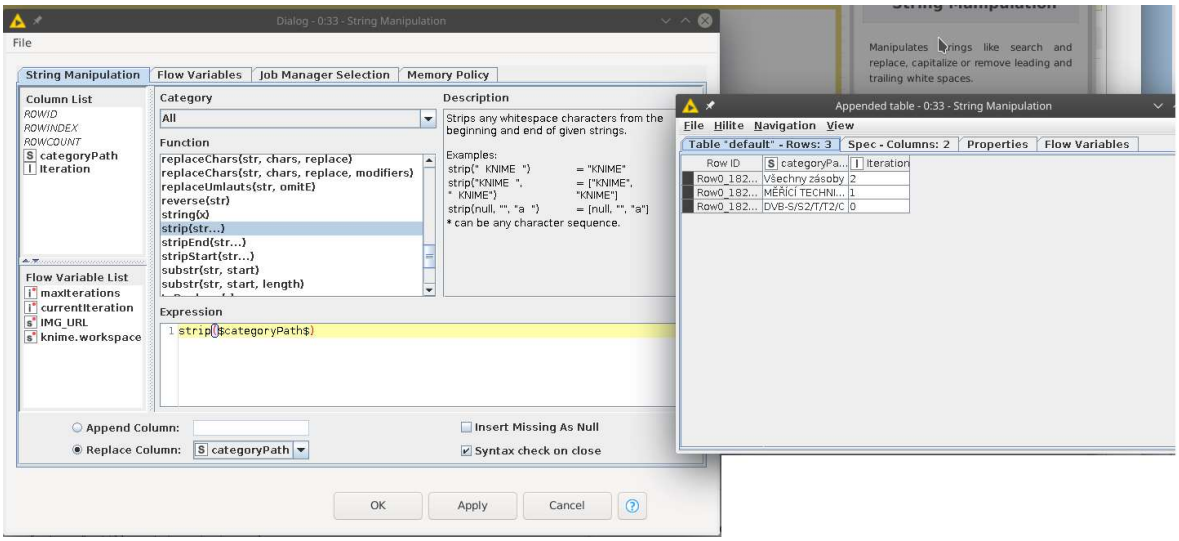
Obrázek 15 - výsledek před a po rekurzním cyklu

Stále s transformací těchto dat toto není vše. Kategorie jsou seřazené špatně. Je tedy jejich pořadí obrátit od konce. Díky pořadí iterace existuje řádek, díky kterému lze kategorie obráceně seřadit. K seřazení tabulky se použije Node "Sorter". Sorter umí tabulku seřadit podle jednoho nebo i více sloupců. V tomto případě existuje jen jeden sloupec, podle kterého lze řadit. Řazení se pak v případě více sloupců řeší nejdřív podle prvního sloupce, pak dalšího až k poslednímu.



Obrázek 16 - použití sorter node

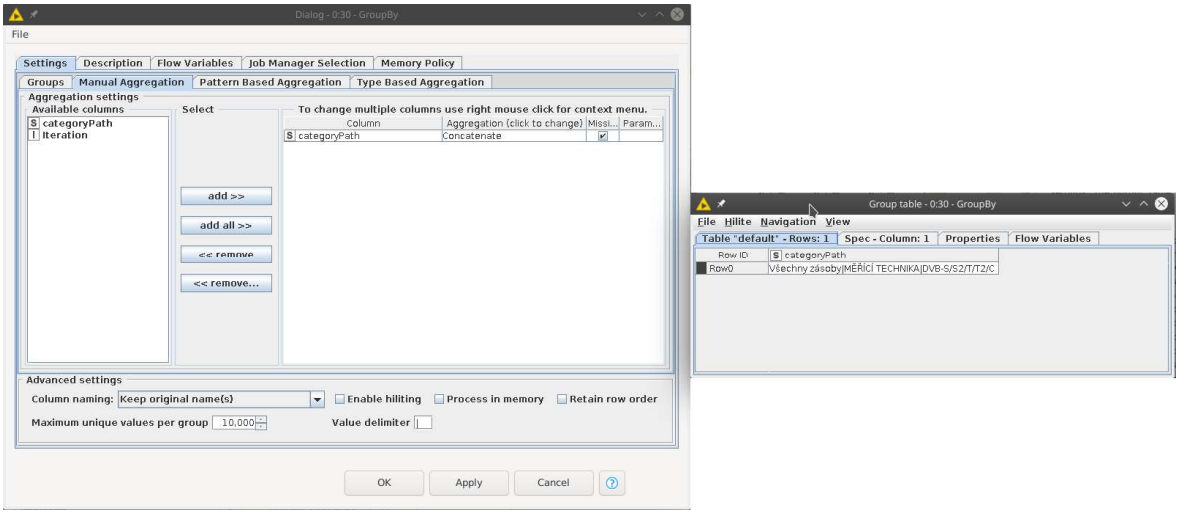
Protože soubor obsahuje chyby v podobě výše zmíněných bílých znaků, je potřeba ještě použít Node String Manipulation k odstranění těchto zbytečných znaků před a na konci názvu kategorie.



Obrázek 17 - použití funkce strip na odstranění bílých znaků

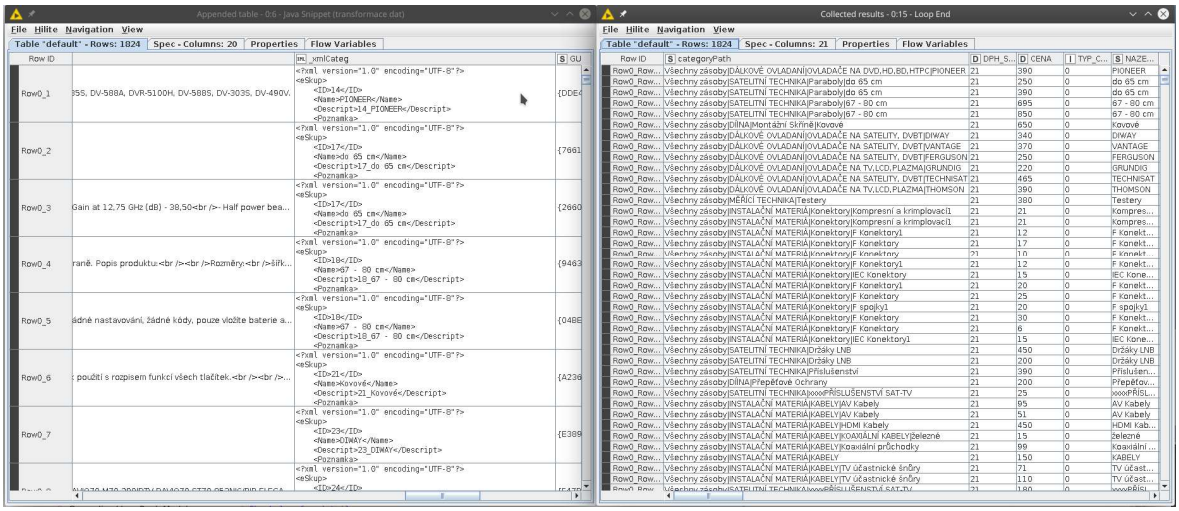
Teď už jen je potřeba všechny řádky sloučit do jednoho řádku s jediným sloupcem, kde hodnoty jednotlivých řádků budou odděleny znakem "|". K tomu pomůže Node GroupBy. Jeho použití nebude až tak zcela podle standardů Knime korektní, ale vše proběhne tak jak má, protože sloupec podle kterého se mají řádky seskupit neexistuje. Aby se předešlo zbytečnému značení chyby, lze tak přidat statický sloupec s libovolnou hodnotou. V nastavení GroupBy stačí jen na záložce "Manual Aggregation" vybrat sloupec který s názvy kategorií. Pak si uživatel vybere, jakou agregační funkci chce použít. Je zde sloupec s typem text, takže výběr není tak široký jako například u číselného sloupce. V tomto případě se vybere funkce "Concatenate", která znamená sloučit. V nastavením v dolní části okna je okýnko, kde se dá vyplnit "Value delimiter". Zde se současná hodnota přepíše právě požadovaným znakem "|". Zmáčkne se OK a spustí Node. V Logu se

vypíšou chyby, že není zvolený skupinový sloupec a že tedy seskupí celou tabulku do jednoho řádku. To je v tomto případě v pořádku, to je přesně potřeba. A když se podíváme na výstup GroupBy, vidíme přesně požadovaný řetězec.



Obrázek 18 - konfigurace group by a výstup

Tady už je zpracování v cyklu dostává ke konci. Na začátku bylo zmíněno, že bude odvedena druhá větev s přijatými daty mimo rekurzivní cyklus ke konečnému sloučení s načtením kategorií. Protože XML s kategoriemi už není potřeba, lze tento sloupec odstranit použitím Column Filter. Celková data se pak spojí pomocí Node Cross Joiner, který funguje tím způsobem, že v jedné tabulky ke každému řádku přiřadí řádek z druhé tabulky. V tomto případě se spojí jen jeden řádek s jedním. K tomuto byl první cyklus, aby docházelo ke zpracování jednoho zboží po druhém. Cross Joiner Node nemá žádné parametry nebo nastavení. Za Node Cross Joiner se umístí pak už jen Loop End a zkusí se spustit. Výsledkem je zas zpátky tabulka se zbožím, bez sloupce s XML a místo toho je zde sloupec s kategoriemi pro e-shop.



Obrázek 19 - výsledek před zpracováním XML kategorie a po zpracování cyklem

Po transformaci dat potřebuje finální podoba ještě několik úprav. Cílové API nebude schopné načítat obrázky z lokální cesty. Lokální cestou k souboru rozumějme takovou cestu, která je přístupná pouze pro zdrojový systém, nikoliv pro cílový. Pro API cesta

„y:\moneys3_fotky_eshop\16541-5058567596712.jpg“ je nepoužitelná a soubor nebude na tomto místě pro cílový systém existovat. Programátor systému však vytvořil jednoduché webové rozhraní, na kterém jsou soubory dostupné externě, a API si je bude moci pak z tohoto místa stáhnout. Proč ale taková cesta k souboru není definována rovnou? Aplikace Money S3 o existenci webového rozhraní na poskytování obrázků neví, tato komponenta je od něho oddělená a Money S3 poskytuje cesty k souborům podle svých vlastních definic.

5.2.1.1 Webové rozhraní pro stahování obrázků ze zdrojového systému

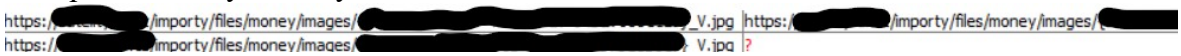
V předchozím odstavci bylo zmíněno rozhraní pro stahování obrázků. Není to žádný seznam dostupných obrázků ke stažení z Money S3. Toto rozhraní předpokládá, že aplikace, která se bude připojovat bude dobře znát přesnou podobu odkazu na obrázek, který bude chtít stáhnout k sobě do cílového systému. Rozhraní je na adrese https://[WEB] /importy/files/money/images/{GUID}_V.jpg a zároveň ještě na https://[WEB] /importy/files/money/images/{GUID}_M.jpg, přistustupuje se metodou ‚GET'. Záměrně z důvodu bezpečnosti, protože jde o systém který je v reálném provozu nebude zde zveřejněna přesná podoba nahrazeného řetězce [WEB]. Protože původní cesta k obrázku zcela neodpovídá požadované podobě odkazu, bude nutné celý odkaz sestrojit znovu. GUID je parametr, který je známý a je součástí exportu. Oba odkazy se odlišují na konci. Jeden odkaz končí _V, druhý _M. Tím se určuje, jestli se má stáhnout obrázek náhledový, nebo obrázek v plné velikosti. GUID je identifikátor zboží, ke kterému obrázek bude pak import hledat.

5.2.1.1.1 Převod URL pro import

Celý tento převod je pouze o dvou krocích. Jeden z kroků je vygeneroval URL na webové rozhraní ze zdrojového systému a to pro náhledový obrázek a zároveň pro obrázek v plné velikosti. Každá položka zboží má ale jiné parametry a ne ke každé položce existují oba obrázky. API pro import dat požaduje, aby odkazy měli formát jednoho řetězce, oddělený znakem |. Pokud existuje jen jeden odkaz, pak má být poskytnut v řetězci pouze jeden odkaz bez dalších oddělovacích znaků.

Import dat zde pomocí dvěma nody typu „String manipulation“. Jejich úkolem je pouze načíst hodnotu ze sloupců OBRAZEK1, druhý node dělá stejnou úpravu se sloupcem OBRAZEK2 s jediným rozdílem a to doplněním _V nebo _M na konec odkazu.

Následuje pak node „Missing Value“, který pracuje dohromady s nodem „Missing Value (Apply)“. Není chyba, že obrázek pro některé položky chybí. Chybou by bylo, kdyby v prázdných řádcích knime doplnil hodnotu NULL. Jako datový typ pro strojové porovnání je tato hodnota v pořádku, pro výstup do dalšího systému se tento typ chová jako řetězec a textová hodnota „NULL“ a datový typ NULL jsou naprosto odlišné metody. Missing Value má jen tyto NULL typy zachytit a doplnit prázdným řetězcem. NULL typ je v Knime reprezentovaný červeným otazníkem.

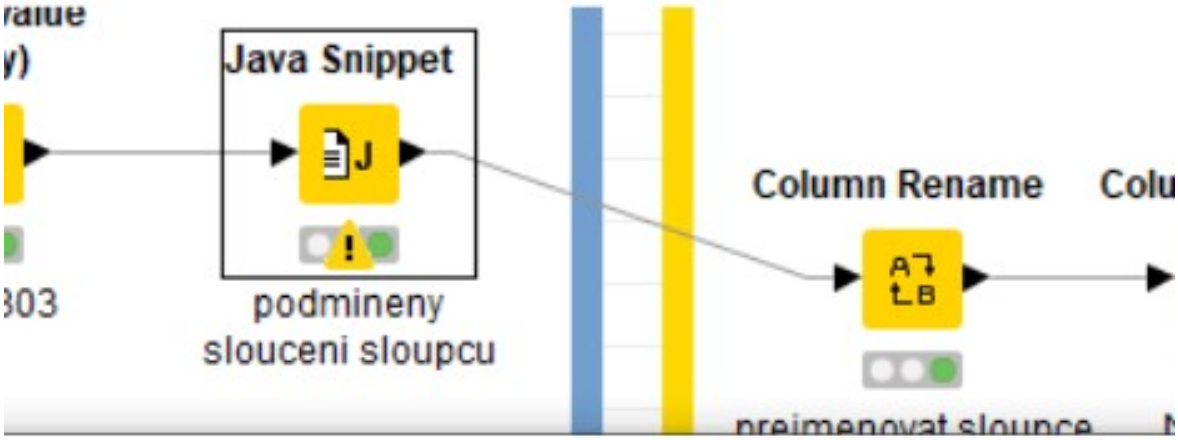


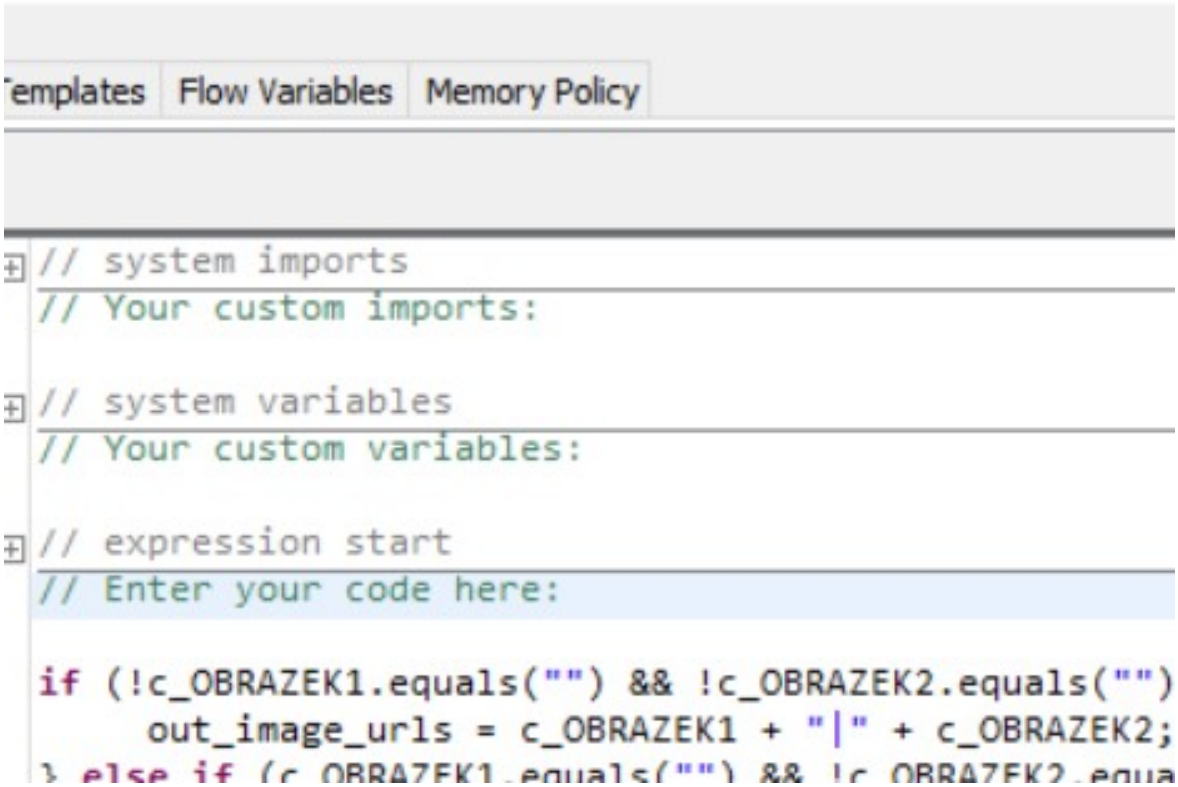
Obrázek 20 - Null v knime nevyhovující importu

Obrázek 21 – po konverzi na prázdný textový řetězec

Celá konverze tímto ale hotová není. Cílový systém a import dat požaduje, aby tato url byla předána v jediném sloupci a formát předaných URL jsou URL oddělená pomocí

znaku ‚|‘, pokud je URL jediná pak stačí předat URL bez oddělujícího znaku. K tomu bylo zvoleno použití jednoduchého kódu, který tyto případy zachytí a vygeneruje správnou hodnotu v požadovaném formátu. Kód bude napsaný v programovacím jazyce Java a ten se dá napsat do node „Java Snippet“.





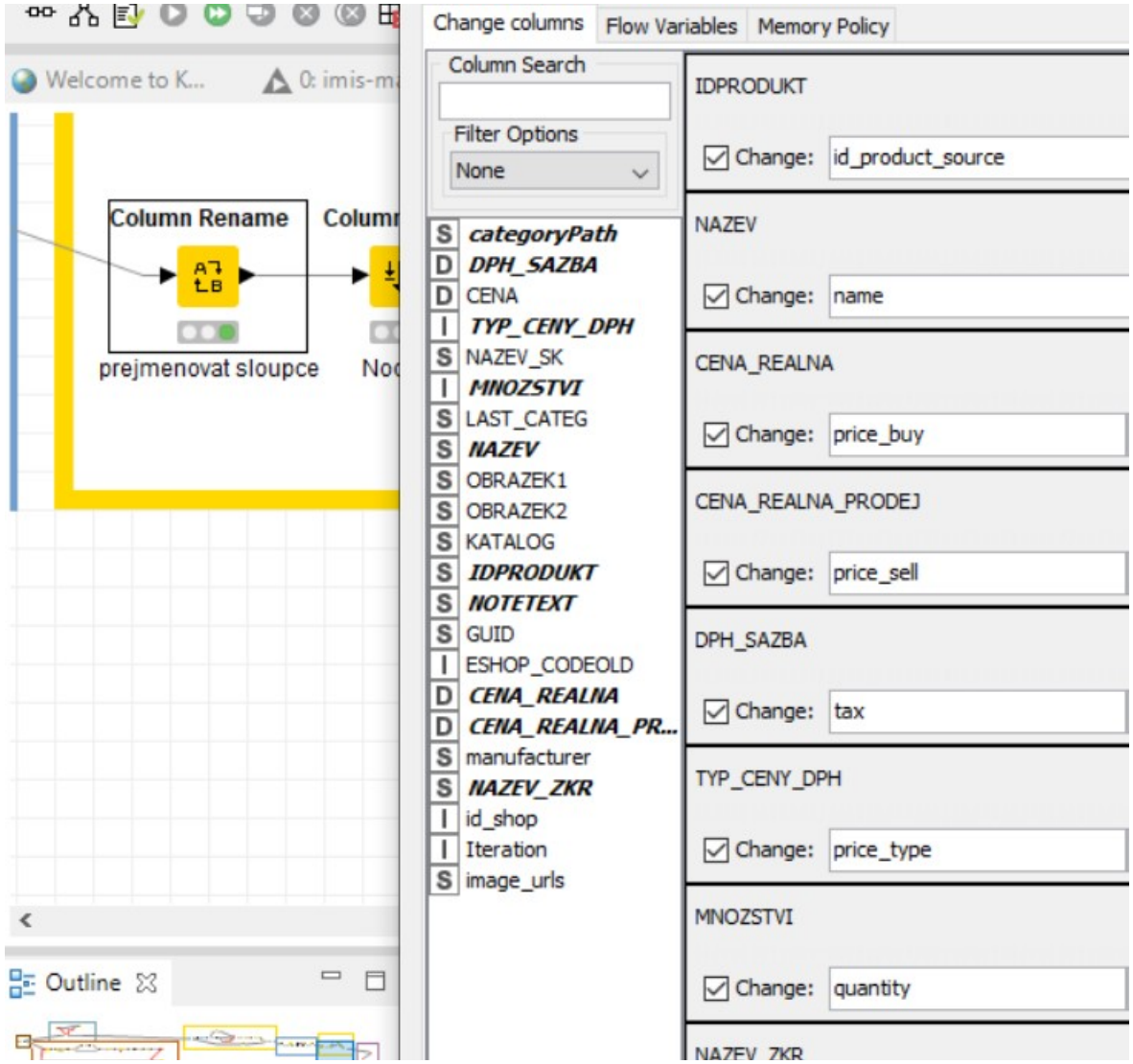
Obrázek 22 - java kod na sloučení sloupců

Výsledkem napsaného zdrojového kódu je prázdná hodnota v řádku, nebo doplněné obě URL oddělující znakem. V posledním případě je doplněná pouze jedna URL bez oddělujícího znaku.

Obrázek 23 - Po konverzi Java kódem

5.3 Import dat

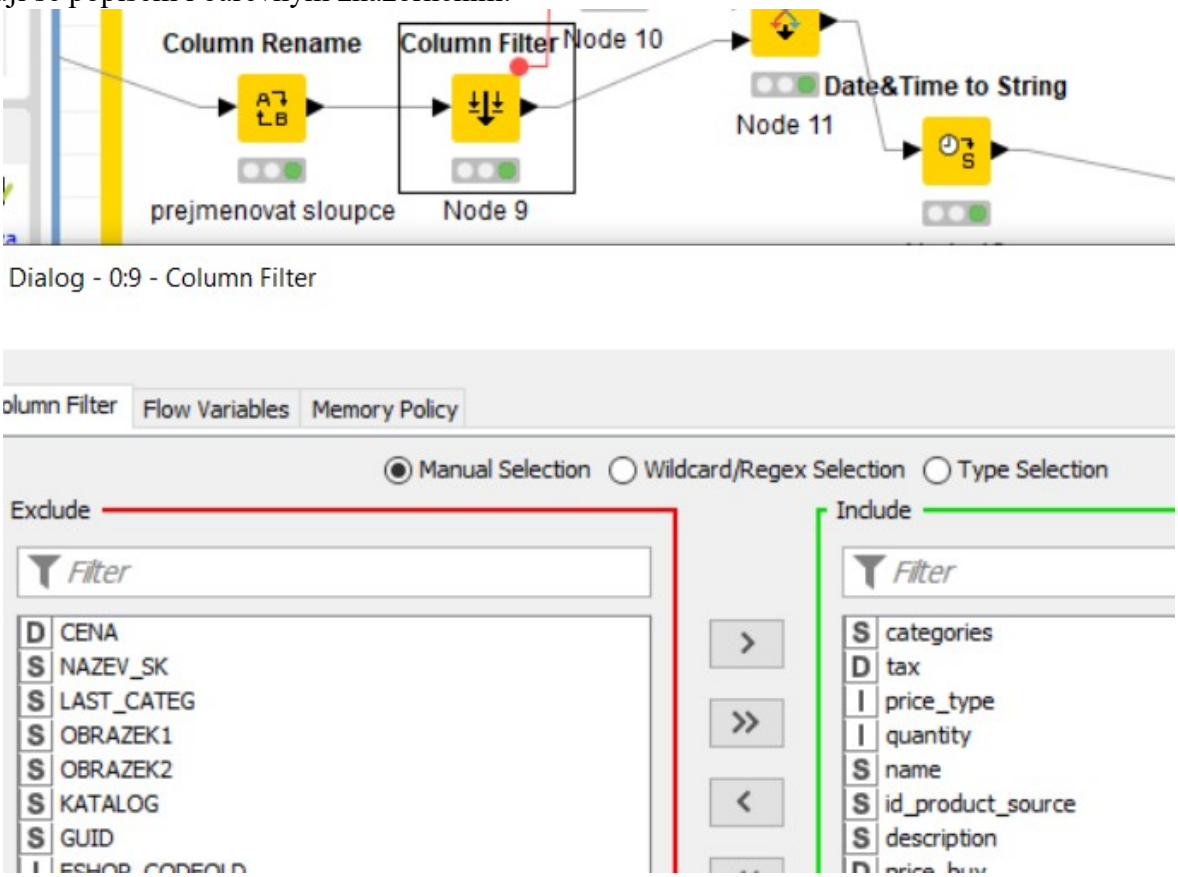
Aby import dat do cílového systému proběhl správně, je nejsnazší ve schématu připravit data tak, aby cílový systém nebyl zatěžován například sloupci, které nejsou požadovány. Před nahráním do databáze se musí udělat poslední transformace. Nejdříve jsou zde sloupce, které se mají do importu přenášet, ale nemají správný název. K tomu pomůže node „Column Rename“. V nastavení tohoto node se nastaví ty sloupce, které se mají přejmenovat a nastaví se tak jejich nový název. Node umožňuje přejmenovat všechny sloupce naráz.



Obrázek 24 - Přejmenování sloupců

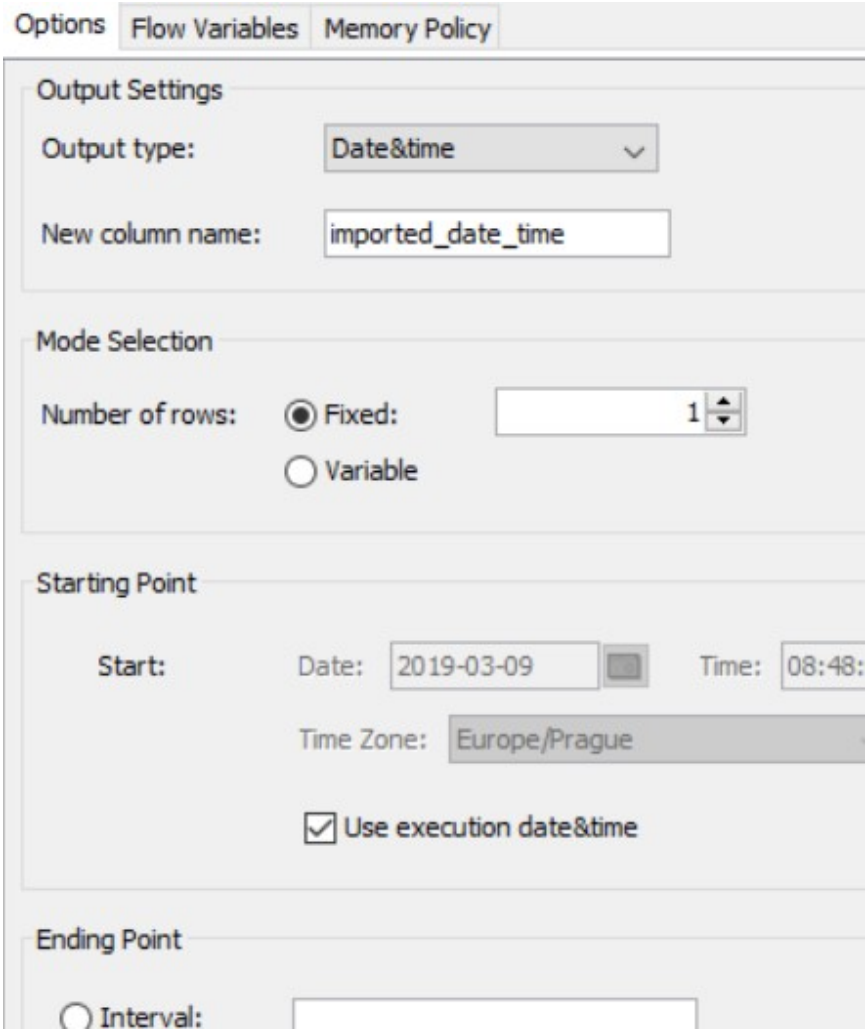
Přejmenování rovněž umožňuje i jednoduchou datovou transformaci. V tomto případě toho nebylo využito, Knime správně vyhodnotil, které sloupce odpovídají kterému datovému typu už podle předchozího typu nebo hodnoty. Tady už se v předchozí krocích datové typy odladili.

Po přejmenování následuje filtrace sloupců. Úkolem filtrace v tomto je odstranit z výstupu ty sloupce, které již nebudou potřeba. Nastavení node „Column Filter“ obsahuje 2 seznamy. Seznam sloupců, které se mají odstranit a seznam sloupců, který má zůstat. Odlišují se popisem i barevným znázorněním.



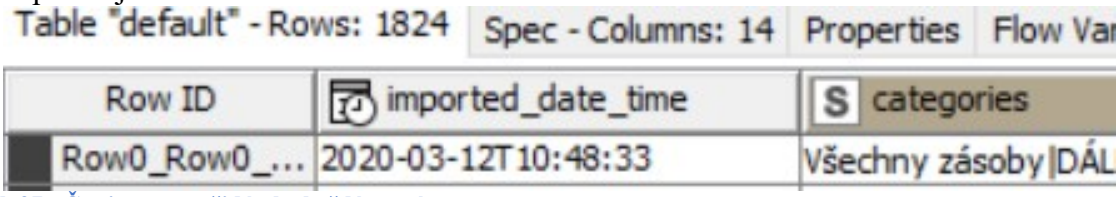
Obrázek 25 - Nastavení filtrace sloupců

Po této transformaci jsou data téměř připravena k nahrání. Jediný údaj který zde chybí je datum importu. Datum importu se dá vygenerovat přímo v Knime. K tomu slouží node „Create Date&Time Range“. Tento node při svém spuštění vygeneruje jeden řádek s požadovanou hodnotou času. Stačí v tomto ohledu jeden datum a čas v době, kdy byl spuštěn import. Tento node se dá nastavit od kdy, do kdy má generovat časové rozmezí a rovněž i nastavit tak interval. Tady se zaškrtne v podokně „Start“ a „End“ hodnota „Use execution date&time“. Protože je požadován jen jeden datum a to doba spuštění, v počtu řádků se nastaví hodnota 1. Vygeneruje se tak jediný řádek s aktuálním datem a časem. Stejně tak v nastavení se i vybere správný název sloupce, v tomto případě „imported_date_time“.



Obrázek 26 - Nastavení data a času

Po vygenerování data je potřeba toto jedno datum přidělit ke každému řádku v celém importu. Tento počin se dá realizovat pomocí node „Cross Joiner“. Ten má za úkol vzít řádky ze 2 tabulek a spojit ke každému řádku jedné tabulky řádek z druhé tabulky. V tomto případě se přidělí 1 řádek s časem ke každému řádku z importu, protože generátor času vytvořil pouze jeden řádek.



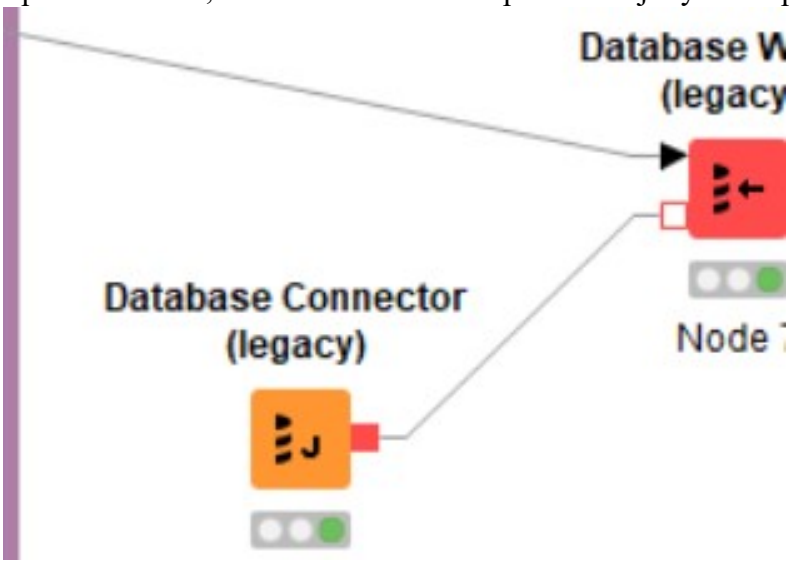
Obrázek 27 - Čas importu přidán ke každému záznamu

Výsledný import dat požaduje správný formát data. Bude proto použit node „Date&Time to String“ dle požadavků cílového systému a jeho databáze na technologii MySQL. (citace https://dev.mysql.com/doc/refman/8.0/en/datetime.html) Předpoklad byl že si technologie Knime samostatně poradí s importem datového záznamu do databáze. Nesčetné pokusy o import date&time formátu do MySQL ukázal, že Knime má i drobné nedostatky. Textový formát datumu do databáze fungoval správně. Proto se převádí sloupec s datem importu do formátu text. Dle zkušeností autora s vývojem tato metoda

funguje i v běžném programování. To potvrzuje i návod například umístěný zde (citace https://www.geeksforgeeks.org/php-date-format-when-inserting-into-datetime-in-mysql/) kde je požadované datum vkládáno do databáze jako text ve správném formátu. Proto bylo usouzeno že tato metoda bude fungovat a následně to bylo i potvrzeno úspěšným vložením do databáze, popisované v následující kapitole.

5.3.1 Připojení k cílovému systému

Poslední krok je nahrát celý import do databáze. K tomu je potřeba stejně jako jinde se do cílové databáze připojit a následně provést nahrání dat. V předchozích kapitolách byl popsán proces načtení, transformace a další úprava zdrojových dat pro finální import.



Obrázek 28 - import transformovaných dat do databáze

V node „Database Connector (legacy)“ se nadefinuje připojení k databázi. Jakmile nastavení projde úspěšně, je možné přes tento node provádět jakoukoliv manipulaci s cílovou databází. Knime umí s databází načítat data, ukládat data, provádět aktualizaci na základě podmínky nebo provést přímo nějaký SQL kód.

V tomto případě bude pouze použit zápis do databáze. A to konkrétně pomocí node „Database Writer (legacy)“. Tento node nemá žádné podmínky, jediné co se zde nastaví je název tabulky, do které se mají data uložit. Node umožňuje přímo nastavit, jestli má zároveň tabulku vytvořit, pokud neexistuje.

5.4 Automatizace importu

5.4.1 Ruční spouštění

Knime jako aplikaci lze spouštět kdykoliv. Hotové schéma importu provede nahrání dat, vždy když se uživateli bude chtít. Uživatel hlavně může prohlížet data krok za krokem od jejich načtení po jejich transformaci až po nahrání do databáze a kontroly stavu.

5.4.2 Automatické spouštění na serveru

Knime lze rovněž nahrát na počítač a nechat ho spouštět vždy v pravidelných intervalech. Při průzkumu diskuzního fora samotného Knime byl nalezen tento návod,

který přesně tento problém řeší (citace https://forum.knime.com/t/run-knime-workflow-on-batch-mode-linux/32293/2). Díky tomuto popisu byl vyzkoušen postup spuštění knime na linux serveru. Byl vytvořen shell skript který provede tento příkaz: *knime -consoleLog -noexit -nosplash -workflowFile="DATOVY_MUSTEK.knwf" -application org.knime.product.KNIME_BATCH_APPLICATION*

Jediné co je nutné vyřešit je formát .knwf který je zde požadován. Standardně je schéma uložené v adresářové struktuře. V tomto formátu schéma nelze spustit pomocí shell skriptu. Konverze do .knwf formátu se provede díky funkce „Export". Tato funkce právě převede schéma do souboru ve formátu .knwf, který je spustitelný skriptem. Automatické spouštění skriptu pak lze realizovat pomocí CRON jobu, což je naplánovaná úloha.

6 Nasazení datového můstku

V současnosti pro přenos dat není pro přenos dat používáno žádné softwarové řešení. Od programátora byla zhotovena pouze částečná příprava pro požadovaný import dat a tím je možno stahovat obrázky do cílového systému. Celý přesun dat je založen čistě na lidské práci, kde majitel si přenáší data ručně za pomoci připraveného skriptu od svého programátora e-shopu.

6.1 Analýza stavu

Cílem analýzy stavu je srovnat aktuální import dat, jeho náročnost a pracnost. Následně zhodnotit přínos jeho automatizace stejně jako jeho cenu.

6.1.1 Současný stav

V současnosti se tyto přenosy realizují ručně. Od programátora e-shopu mám informaci, že pro import existuje jakýsi skript, který umí získaný xml načíst do databáze. Neexistuje zde však vizualizace a jediná zpráva o importu je ta, kterou vygeneruje až pak samotný cílový systém. Pokud tato operace selže, musí přijít programátor a celou situaci řešit. Dle zjištěných informací v průměru tato práce trvá 2 hodiny času programátora, kde se tato informace bude považovat za stěžejní pro zhodnocení současných výdajů. Majitel e-shopu si nepřál uveřejňovat interní informace o výdělku, bude tedy následné zhodnocení čistě teoretické na základě průměrného výdělku podnikatele v Ústeckém kraji, ze kterého pochází i autor této diplomové práce. Majiteli zabere používání současného importního programu od svého programátora přibližně půl hodiny času denně. V případě větších změn provádí import častěji. Bude se posuzovat časová náročnost v průměru na 2 importy denně, tedy majitel e-shopu stráví hodinu času denně importem dat namísto aby tuto hodinu času věnoval své práci a zlepšil své výdělky.

6.1.1.1 Situace v Ústeckém kraji

Podle webové stránky prumerneplaty.cz se hodinová hrubá mzda zaměstnance v ústeckém kraji a konkrétně v Ústí nad Labem pohybuje okolo 31 tisíc Kč. Z poznatků a osobních zkušeností autora se mzda pohybuje spíše kolem 23 tisíc Kč. Pro další kalkulace bude použita hodnota průměrná mezi mzdou udanou dle zdroje a osobních zkušeností autora, a to konkrétně 27 tisíc Kč. Autor diplomové práce tento odhad na základě

získaných informací a osobních zkušeností považuje více za věrohodný. Při práci na plný úvazek dle webové této webové stránky je průměrná hodinová sazba 175 Kč, s ohledem na přepočet dle osobních zkušeností a zprůměrování rozdílu mezi zkušenostmi autora a uvedenou informací se průměrná hodinová sazba pohybuje okolo 152 Kč. Pro zhodnocení současných nákladů se bude předpokládat hodinová sazba IT programátora a hodina práce majitele e-shopu. Předpoklad na základě získaných informací bude vypadat následovně:

Pozice	Hrubá hodinová sazba
IT programátor	152 Kč
Podnikatel v oblasti prodeje elektrotechniky	175 Kč

Tabulka 1 - Srovnání nákladů za mzdu

Jestliže se současný program porouchá při importu a musí zasáhnout programátor za cenu 2 hodin v průměru na opravu problému, stojí tato řešení poruchy jednorázově 304 Kč. Hodina práce majitele e-shopu ho denně stojí 175 Kč, které by mohl vydělat svojí prací. Měsíc má v průměru 31 dní, tedy mohl by vydělat dalších 5 425 Kč.

6.1.2 Navrhovaný stav

Navrhovaný stav počítá s kompletní automatizací celého procesu importu dat. Majitel e-shopu tedy bude schopen se celou svoji pracovní dobu věnovat výhradně svému e-shopu a obchodu. Jediné co v tomhle ohledu bude vyžadovat pozornost je oprava případných chyb automatického importu, pokud dojde k nečekané události. To je stav, který se dodnes nezmění, protože nadále má k dispozici vlastního programátora.

Pro realizaci automatizace bude majitel e-shopu muset najít místo, kde bude provozovat aplikaci Knime a zároveň bude na tomto místě provozovat navržené schéma. Zmíněným místem se jedná o osobní počítač nebo server, kde lze schéma spouštět. Majitel e-shopu samozřejmě má k dispozici server, kde má uložený e-shop který provozuje. Při konzultaci s programátorem bylo zjištěno, že majitel e-shopu provozuje počítač, na kterém lze podobné aplikace spouštět a zároveň jako správce má plný přístup do tohoto počítač. Proto bylo dohodnuto, že tento počítač bude pro automatizaci určen.

6.1.2.1 Současný stav implementace ve společnosti

V době psaní této práce bylo, vyzkoušena implementace na virtuálním počítači autora práce v testovacím prostředí. Plná automatizace byla funkční v testovacím prostředí.

Dalším krokem bude reálná duplikace nastavení testovacího prostředí na počítač majitele a spustit plnou automatizaci. Podrobnější detaily budou následovat v další kapitole.

6.2 Zhodnocení implementace datového můstku

Ve zhodnocení bude popsán detailně současný stav implementace automatizace Knime s navrhovaným schématem a jeho cenová kalkulace. Dále zde budou navrženy i další možnosti nasazení ač dle zadání programátora je současné řešení známé.

6.2.1 Způsoby řešení

K automatizaci je jednoznačně nutné mít počítač, který bude aplikaci Knime provozovat spolu s navrženým schématem importu. V tomto ohledu nelze vymýšlet jiné řešení. Jediné srovnání, které se zde nabízí je použití vlastního počítače nebo si pronajmout server a platit za něj nájem.

6.2.1.1 Vlastní PC

Vlastní PC má výhodu, že neplatí majitel měsíční poplatky vyjma poplatku za elektřinu, kterou počítač spotřebuje. Pro jeho provoz ale musí majitel mít dostatečné znalosti na jeho správu. Jakákoliv podpora v případě problémů, v tomto ohledu neexistuje. Majitel si pronajímá pro potřeby správy PC svého programátora, za jehož práci musí zaplatit. Protože Knime je pod svobodnou licencí, viz. kapitola 2.3.1. Licencování, není nutné platit za samotnou aplikaci Knime. Správa nad touto aplikací musí být ale předána do rukou správce, který s aplikací Knime umí pracovat. Pokud se počítač poškodí, musí sám majitel e-shopu řešit jeho náhradu a znovu zprovoznění automatizace. Cenová politika je v tomto ohledu výhodná pouze za podmínky, že majitel je zdatný v oboru IT.

6.2.1.2 Pronájem serveru

Pronájem serveru nabízí zásadní výhodu. Provozovatel serveru poskytuje záruku na provoz, tedy nemůže dojít k přerušení služeb. Nevýhodou je měsíční platba za jeho provoz. Navíc na provoz Knime je potřeba najít takový server, kde majitel může nainstalovat vlastní aplikaci a spouštět ji v pravidelných intervalech. Taková služba už není tolik často samozřejmostí. Průzkum trhu na internetu ukázal na tyto poskytovatele, dle webu http://www.hostingworld.cz/cz/poskytovatele.html :

Poskytovatel	Služba
WEDOS	VPS-ON
FORPSI	CLASSIC-VPS
Active24	Aplikační servery (APS)
Czechia	Cloud Server VPS

Tabulka 2 - Seznam poskytovatelů

Seznam je podstatně delší, byli vybráni na základě průzkumu známější poskytovatelé serverů. Už podle názvu služeb se jedná převážně o poskytování virtuálních serverů. To není žádná novinka, v současné době je tento přístup moderní.

6.2.2 Cenová kalkulace

Použití automatizace znamená v prvopočátku, než se zaplatí za server nebo připraví počítač, na kterém Knime bude v provozu nejprve návrh samotného importu. To se skládá ze samotné analýzy požadavků, vytvoření schématu a jeho testování. Zhodnocení časových požadavků bude zhodnoceno na základě doby, po kterou trvalo vytvořit schéma importu prezentovaný v této diplomové práci.

Aktivita	Časová náročnost (v hodinách)
Zpracování zadání od programátora	6
Analýza dat	5
Příprava schématu	68
Testování	25
Nasazení do reálného provozu	2

Tabulka 3 - Časová náročnost implementace

Celková doba strávená nad vypracováním schématu trvala autorovi práce 106 hodin. Implementátor Datového můstku by byl hodnocen jako byl programátor v kapitole 6.1.1.1 Situace v Ústeckém kraji. Celková hodnota vývoje Datového můstku pomocí Knime by byla 16 112 Kč. Tato investice je jednorázová. Další investice budou popsány ve vlastních kapitolách.

6.2.2.1 Vlastní PC

V tomto ohledu majitel e-shopu musí pořídit počítač, který bude aplikaci Knime provozovat. Díky hardwarovým nárokům aplikace Knime pro Datový můstek stačí počítač nižšího výkonu.

Na obchodech Alza.cz a CZC.cz, které jsou na území České republiky jedním z nejrozšířenějších byly vybírány počítače s následujícími hardwarovými nároky: 8 GB ram, procesor bez specifických požadavků, grafická bez specifických požadavků, diskový prostor minimálně 240 GB. Protože Knime bude zpracovávat větší objemy dat, je nutné i z důvodu případné výkonové rezervy zvolit minimální dostatečnou paměťovou kapacitu. Na procesoru už tolik nezáleželo, vytvořené schéma nebylo náročné na výkon procesoru. Grafický výkon Knime nevyužívá vůbec. Toto ukázalo testování použití Knime v testovacím prostředí. Obchod CZC.CZ nabízí takové počítače v cenové relaci od 8 990 Kč, obchod Alza.cz od 6 490 Kč.

S ohledem na porovnání konkurence se výhodněji jeví počítač z obchodu Alza.cz. Pro zjednodušení režie nad počítačem v rámci testování se osvědčil OS Debian. Reinstalace PC trvá dle zkušeností autora hodinu až dvě podle výkonu PC. Nainstalování Knime je jen záležitost doby stahování softwaru ze stránek autora aplikace. Nahrání schématu a spuštění skriptu se hodnotí na dobu trvání jedné hodiny. Při průzkumu servisních nabídek na PC se ceny firem pohybují v rozmezí od 400 až po 700 Kč. Na základě průzkumu se bude předpokládat cenová hodnota 550 Kč, což je průměr zjištěného rozmezí. Pokud tedy nezvládne uživatel sám nainstalovat počítač sám, bude ho tato služba stát další peníze v hodnotě od 550 Kč až po 1100 Kč. Bude se předpokládat dražší varianta, dle zkušeností autora si linux instalují nejčastěji samotní správci, tedy osoby které se systémem umí pracovat a nejsou standardní uživatelé.

Celková investice do vlastního PC je jednorázová, tedy **7 590 Kč**. Následně musí mít majitel e-shopu připojení k internetu, které se předpokládá, že má již zřízené, protože už nyní provozuje e-shop. Jediné co majitele e-shopu bude tížit je cena za energie, kterou PC spotřebuje.

Pokud dojde na automatizaci, při dalším zisku majitele e-shopu 5 425 Kč, který byl vypočítán v předchozí kapitole se tato investice navrátí celkem za 1,4 měsíce. Nevýhodou je, že jakákoliv porucha na tomto počítači znamená nutnost najmout IT služby další firmy, které opravy zvládnou.

6.2.2.2 Pronájem serveru

Dle průzkumu v předchozích kapitolách se ceny pronájmu serveru pohybují od 45 Kč za měsíc až po 990 Kč za měsíc. V průměru se jedná o 517 Kč, který by byl měsíčním výdajem. Výhodou je že všechny úkony správce padají na poskytovatele serveru a majitel e-shopu nemusí mít s tímto žádné starosti. Jen musí v rámci jednorázové investice zajistit, že bude aplikace na serveru nahrána a spuštěna do provozu.

6.2.2.3 Shrnutí

Záměrně bude zhodnoceno nasazení na server nebo vlastní PC v této kapitole, protože se nijak v obou případech neliší a na základě poznatků bylo v této diplomové práci zhodnoceno na dobu 2 hodin práce. Dle předchozích průzkumů a zhodnocení je stanovena tato práce na hodnotu 304 Kč, protože nasazení by prováděl programátor majitele e-shopu a byl hodnocen dle předchozí tabulky. 304 Kč by byla první jednorázová investice. Při pořízení vlastního PC se jedná o investici 7 590 Kč, která by se vrátila za 1,4 měsíce. Pokud díky automatizaci majitel vydělá díky své práci dalších 5 420 Kč měsíčně, investice do vlastního PC se vrátí za 1,5 měsíce a tato investice je jednorázová. Při zvolení varianty pronájmu serveru automatizací sice získá dalších 5 420 Kč, ale přijde pronájmem serveru o 550 Kč, tedy by nezískal měsíčně 5 420 Kč ale jen 4 870 Kč.

Pak je zde investice za samotné vytvoření datového můstku a dle výpočtu se tato investice vrátí za tři měsíce po spuštění plné automatizace. Při zvolení varianty s pronájmem serveru je doba návratnosti investice 4 měsíce.

6.2.3 Přínos datového můstku

Při celkovém zhodnocení implementace datového můstku přinese ve všech případech peněžní výhodu. Po návratu investice začne vynášet za nejdříve 4 měsíce.

Další zásadní výhodou je grafická aplikace. Protože Knime kromě toho že se dá provozovat na serveru ke spuštění přenosu dat dle vytvořeného schématu, lze na toto schéma koukat pohledem obyčejného uživatele a pracovat s touto aplikací lze naučit i běžné uživatele PC v režimu náhledu a sledování celého procesu transformace dat.

Největší výhodou je licenční politika softwaru. Samotný Knime v základní verzi, která poskytuje nadmíru potřebných funkcí je k dispozici zdarma.

Návrh složitějších procesů ve srovnání s programováním čistě aplikace v rámci časové náročnosti je nesrovnatelně jednodušší právě v aplikaci Knime.

6.2.4 Nevýhody použití Knime

Nevýhoda v tomto ohledu je, že implementaci na server už musí provést uživatel PC na úrovní expert nebo přímo IT technik se znalostí skriptovacích jazyků.

Dále, návrh schématu, kde je i porovnáno použití části transformace pomocí node „Java Snippet“ nebo většího bloku nodů bez znalosti programovacího jazyka je záležitostí zkušenější uživatele se schopností analytického myšlení. Při návrhu tohoto Datového můstku se ukázalo, že zkušenosti s vývojem softwaru a managementem databázových systémů jsou značnou výhodou.

7 Závěr

V této diplomové práci byl navržen Datový můstek pomocí Knime Analytics platform, který zvládl přenést data ze zdrojového systému do systému cílového. Během návrhu bylo prokázáno a autor práce poukazuje na skutečnost, že návrh pomocí této aplikace a zároveň návrh takového můstku pomocí programovacího jazyka je v rámci časové náročnosti až nesrovnatelný. Návrh v aplikaci Knime je podstatně snazší a hlavně už při návrhu schématu mohl autor práce sledovat přímo transformaci dat tak, jak je. Při návrhu vlastní aplikace je takové ladění složitější a vyžadoval by i další specializovaný software pro usnadnění práce. Zároveň po návrhu Datového můstku bylo zhodnocena ekonomická náročnost na tento projekt, se zaměřením na Ústecký kraj ze kterého pochází autor diplomové práce. Toto porovnání bylo snazší, protože autor prostředí Ústeckého kraje zná a mohl posoudit lépe informace získané díky internetu proti osobním zkušenostem a zkušenostem lidí ze svého okolí. Cenová náročnost na implementaci této bakalářské práce byla stanovena na čtvrt roku a to pouze teoreticky, protože s reálnými hodnotami nebylo možné pracovat, jelikož majitel e-shopu si tyto hodnoty nepřál zveřejnit. Datový můstek je v současnosti připraven k implementaci a právě je domluveno přes programátora e-shopu, že tento Datový můstek bude provozován na vlastním PC majitele e-shopu.

Součástí příloh je i vytvořené schéma ale pouze ve formátu PDF. Knime si ukládá hesla v textové podobě a zveřejnit takto nastavené schéma by bylo ohrožením bezpečnosti sítě majitele e-shopu.

8 Seznam použitých zdrojů

1. Knime Analytics Platform. *KNIME.* [Online] 2004. https://knime.com/.
2. Forum Knime Analytics Platform. *FORUM KNIME.* [Online] 2004. https://forum.knime.com/.

9 Přílohy

Datovy_mustek_PDF.pdf