TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 / Elektrotechnika a informatika
Studijní obor: 1802R022 / Informatika a logistika

Automatizace konverze datových formátů pro databázový systém

Automatization of conversion of data formats for a database systém

Bakalářská práce

Autor: **David Krejbich**
Vedoucí práce: Ing. Jakub Říha
Konzultant: Mgr. Kamil Nešetřil

V Liberci 17. 5. 2013

Místo této strany je vloženo originální zadání bakalářské práce

1. Seznámení se s formáty vstupních dat (Labsystém, ČGS-Geofond, případně další) a s rozšířeným formátem aplikace EnviroInsite.
2. Volba vhodného svobodného nástroje Spatial ETL (Extract-Transform-Load) a konverze mezi datovými formáty s jeho použitím.
3. Návrh a realizace samostatné aplikace pro automatický převod mezi některými datovými formáty.
4. Zhodnocení a porovnání užitých přístupů.

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

V Liberci dne 17. května 2013

…….
David KREJBICH

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Jakubu Říhovi a konzultantu Mgr. Kamilu Nešetřilovi za poskytnutí odborných rad, věcné připomínky, ochotu a vstřícný přístup během zpracování této práce.

Velké poděkování náleží celé mé rodině a přítelkyni za podporu, trpělivost a povzbuzování po dobu mého studia.

Abstrakt

Cílem bakalářské práce je porovnání a zhodnocení dvou různých přístupů pro konverzi datových formátů. Prvním z nich je vybrání vhodného open source nástroje ETL (extract, transform, load) a následná konverze datových formátů s jeho použitím. Druhým přístupem je vytvoření vlastní aplikace v preferovaném programovacím jazyce a konverze mezi vybranými datovými formáty.

Vstupními daty jsou vybraná data získaná z databáze řízenou informačním a řídícím programem Labsystém, a vybraná data z archivu Geofond České geologické služby. Cílovým formátem je rozšířený datový model aplikace EnviroInsite. K převedení vstupních dat do cílového formátu je nejprve nutné seznámit se s těmito formáty, což je první částí této bakalářské práce.

V druhé části je popisován převod dat za pomoci vlastní aplikace a za pomoci vybraného nástroje ETL. Zároveň jsou zde popsány známé problémy a omezení těchto aplikací.

Poslední část se zabývá hlavním cílem této bakalářské práce. Tím je zhodnocení a porovnání použitých přístupů pro konverzi datových formátů. Součástí bakalářské práce jsou přílohy, ve kterých jsou uvedeny uživatelské návody k oběma aplikacím.

Klíčová slova

Pentaho Data Integration, Kettle, extract, transform, load, Java, transformace dat

Abstract

The aim of this bachelor thesis is to compare and to evaluate two different approaches for the conversion of data formats. The first one is a selection of an appropriate open source ETL tool (extract, transform, load) and the subsequent conversion of data formats with its use. The second one is a creation of a custom application in preferred programming language and a conversion between the selected data formats.

The input data are selected and obtained from the database managed by Labsystém software and from the archive of Geofond (Czech geological survey). The target format is an extended data model of an application EnviroInsite. To convert input data into the target format, it is necessary to become familiar with their format, which is the first part of this thesis.

In the second part of this bachelor thesis, the data transfers using the created custom application and the selected ETL tool are described. At the same time, known issues and limitations of those applications are described.

The last part deals with the main aim of this bachelor thesis which is the evaluation and the comparison of used methods for the conversion of data formats. An addendum containing user guides for both applications is also part of this thesis.

Key Words

Pentaho Data Integration, Kettle, extract, transform, load, Java, data transformation

Obsah

Seznam tabulek

Seznam obrázků

Seznam zkratek

TUL	Technická univerzita v Liberci
ETL	Extract, transform, load
IS MARE	Informační systém MARE
DBF	dBase Table File Format
PDI	Pentaho Data Integration
SQL	Structured Query Language
BI	Business Intelligence
ODBC	Open Database Connectivity

Úvod

Tato bakalářská práce se zabývá konverzí datových formátů za pomoci vybraného nástroje ETL (Extract, Transform, Load) a za pomoci vlastní aplikace. Cílem je porovnání těchto dvou užitých přístupů. Práce se rozděluje na část teoretickou a na část praktickou. V teoretické části jde především o seznámení se se vstupními formáty dat a s rozšířeným formátem aplikace EnviroInsite, který slouží jako formát cílový. Praktická část se zabývá prací s vybraným nástrojem ETL a tvorbou vlastní aplikace.

V případě vstupních formátů se jedná o vybraná data z archivu Geofond, který je složkou státní organizace Česká geologická služba, a o data z databáze, kterou využívá organizace AQUATEST, a. s.

Cílovým formátem je rozšířený datový model aplikace EnviroInsite, která načítá soubory ve formátu Microsoft Access nebo Microsoft Excel. Aplikace EnviroInsite, sloužící pro vizualizaci hydrogeologických dat, umožňuje rozšiřovat svůj formát v podobě načítání vlastních dat. Toho využívá Informační systém MARE, vyvíjený v rámci projektu MARE na Technické univerzitě v Liberci. IS MARE slouží mimo jiné pro správu bodových měření a informací o vrtech.

Hlavní část bakalářské práce je věnována výběru vhodného nástroje ETL, konverzi mezi datovými formáty s jeho použitím a tvorbě vlastní aplikace pro vybraný převod.

Závěrečné kapitoly práce jsou zaměřeny na popis, zhodnocení a porovnání užitých přístupů. Součástí závěru je výběr vhodnějšího přístupu pro konverzi mezi datovými formáty.

1 Vstupní a výstupní formáty dat

Vstupní data byla získána od organizace AQUATEST, a. s. a od České geologické služby – Geofondu. V obou případech se jedná o několik souborů, které mají různé formáty a různou strukturu uložených dat. Seznámení se s těmito soubory a s formátem jejich dat je nutnou podmínkou k tomu, aby bylo možné tato data transformovat do požadovaného formátu. V následujících kapitolách jsou popsány formáty vstupních dat (rozděleny dle organizací, která data poskytla) a rozšířený formát aplikace EnviroInsite.

1.1 AQUATEST a. s.

AQUATEST, a. s., poskytuje konzultantské a inženýrské služby v oblastech životního prostředí a vodního hospodářství. Pro zpracování výsledků laboratorních rozborů, jejich archivaci a vyhotovení podkladů využívá vlastní laboratorní databázi řízenou informačním a řídícím programem Labsystém. [1]

Právě z této databáze pocházejí soubory, které jsou v bakalářské práci používány jako vstupní soubory pro některé transformace.

1.1.1 Soubory z databáze Labsystém

Jedná se o dva soubory ve formátu dBase (přípona *dbf*), což je výchozí formát prvního masově rozšířeného systému řízení báze dat. Tento zastaralý formát je využívaný v mnoha aplikacích, kde je třeba uchovávat strukturovaná data v jednoduchém formátu.

První soubor se jmenuje *hodnoty.dbf* a nese data o měřených veličinách a jejich hodnotách. Tabulka uložená v souboru obsahuje 8 sloupců s daty. Kromě názvu měřené veličiny, naměřené hodnoty, jednotky této hodnoty a metody měření, má každý řádek ještě jeden podstatný údaj, kterým je číslo rozboru. Toto číslo slouží jako klíč pro spojení záznamů této tabulky se záznamy z tabulky druhého souboru.

Ten je pojmenovaný *vzorky.dbf* a obsahuje data o jednotlivých měřeních v určitých lokalitách. Jedná o tabulku s daty uloženými v sedmi sloupcích. Na každém řádku je mimo jiné uvedeno místo, kde se provádělo měření, kdo a kdy prováděl měření a číslo rozboru. Pomocí tohoto čísla můžeme snadno získat informace o tom, jaké veličiny (ze souboru *hodnoty.dbf*) byly v daných objektech (*vzorky.dbf*) měřeny.

Jednotlivé záznamy v obou tabulkách jsou uloženy prostřednictvím různých datových typů, které je třeba při transformacích dat zohlednit.

1.2 Česká geologická služba – Geofond

Česká geologická služba se zabývá sběrem a zpracováním údajů o geologickém složení území České republiky. Tato data předává státním orgánům pro hospodářskou, politickou a ekologickou správu. Navíc poskytuje všem zájemcům regionální geologické informace. Vstupní data byla získána z archivu Geofond spadajícího pod ČGS. [2]

1.2.1 Formát souborů z archivu Geofond

Jedná se o tři soubory, vázající se k určité lokalitě. Jsou uloženy s příponou *mdb*, užívanou pro formát souborů aplikace Microsoft Access od firmy Microsoft. Tato aplikace slouží pro práci s relačními databázemi. Každý ze souborů obsahuje několik tabulek s daty.

Soubor *vrty.mdb* obsahuje informace o vrtech, soubor *profily.mdb* nese informace o tom, v jaké hloubce bylo měření prováděno, a soubor *lokalita.mdb* obsahuje záznamy o měřeních v jednotlivých profilech. Ke každému záznamu jednotlivých měření, profilů či údajů o měřeních se váže číselný údaj označený jako *klic_gdo*, pomocí kterého je možné záznamy těchto souborů spojit a tím přiřadit k sobě informace, jenž spolu souvisí.

Data jsou uložena v souborech v různých formátech datových typů. Některé záznamy, nesoucí číselné hodnoty, jsou uloženy jako textový řetězec, zatímco v cílovém formátu je vyžadován pro tyto záznamy datový typ double. Jiné záznamy obsahují na začátcích či na koncích řetězců nechtěné **bílé znaky**[1], takže délka pole převyšuje počet znaků samotného záznamu.

[1] **Bílý znak** je v informatice takový znak, který představuje prázdné místo neboli mezeru. Typickými zástupci bílých znaků jsou znaky zadávané na klávesnici mezerníkem a tabulátorem. [3]

1.2.2 Soubor vrty.mdb

V této databázi je uloženo 10 tabulek s daty o vrtech. Tabulka, která obsahuje záznamy o vrtech, se jmenuje *vrty*. Některé informace ovšem popisuje pouze zkratkami, jejichž významy jsou rozepsány v ostatních tabulkách.

V následující tabulce je přehledný soupis všech tabulek tohoto souboru s jejich popisem.

Tab. 1: Popis datových tabulek v souboru vrty.mdb

Název tabulky	Popis
KOD_ALL_ORG_R	Kódy a názvy organizací
KOD_ALL_STRAT_R	Kódy a názvy stratigrafických jednotek
KOD_ALL_DRUHOBJ_R	Kódy a názvy druhů objektů
KOD_ALL_UCELOBJ_R	Kódy a názvy účelů objektů
KOD_ALL_ZAMXY_R	Kódy a významy informací o způsobu stanovení souřadnic objektu
KOD_ALL_ZAMZ_R	Kódy a významy výškových systémů
KOD_ALL_DRUHHL_R	Kódy a významy druhů hladin podzemní vody
KOD_ALL_ZK_R	Kódy a názvy realizovaných zkoušek a měření
vrty	Záznamy o vrtech
vrty_E	Popis významů jednotlivých záznamů v tabulce vrty

1.2.3 Soubor profily.mdb

Soubor *profily.mdb* obsahuje záznamy o jednotlivých geologických vrstvách, které dohromady vytvářejí geologický profil vrtu. V souboru jsou uloženy dvě tabulky. První z nich nese informace o geologických vrstvách vrtů a druhá plní funkci popisu jednotlivých záznamů z první tabulky.

Každý vrt je rozdělen do několika řádků podle rozsahu hodnot, ve kterých bylo měření prováděno. Hodnoty tohoto rozsahu jsou uloženy ve dvou sloupcích. V prvním sloupci, pojmenovaném *METR_OD*, jsou uloženy hodnoty v metrech, které určují hloubku, od které v daném profilu probíhalo měření. Ve druhém sloupci, pojmenovaném *METR_DO*, jsou uloženy údaje v metrech, které určují, do jaké hloubky měření probíhalo.

1.2.4 Soubor lokalita.mdb

Jedná se o nejkomplexnější soubor dat z archivu Geofond. Obsahuje 13 tabulek, z nichž jen některé nesou záznamy o naměřených hodnotách týkajících se podzemní vody. Datové tabulky jsou popsány v následující tabulce.

Tab. 2: Popis datových tabulek v souboru lokalita.mdb

Název tabulky	Popis
ASGI	Záznamy o zprávách a posudcích uložených v archivu ČGS – Geofondu
CERP	Záznamy o měření
GEOFOND_H	Kódy zkratek a hodnoty jejich významů
HLAD	Data o hladinách podzemní vody
CHEM_1	Záznamy o chemických rozborech
CHEM_2	Záznamy o chemických rozborech
CHEM_3	Záznamy o chemických rozborech
CHEM_5	Záznamy o chemických rozborech
CHEM_6	Záznamy o chemických rozborech
INTERVALY	Hloubkové intervaly, na kterých probíhalo měření
TEPL	Měření teploty
ZUO	Popis všech objektů, kde se měření provádělo
ZUO_P	Přiřazení posudků k jednotlivým objektům

1.3 Aplikace EnviroInsite

K současnému grafickému zobrazení geologických a geochemických dat existuje řada komerčních počítačových programů, avšak tyto programy jsou vesměs velmi drahé a náročné na obsluhu. Jedinou nám známou výjimkou je software EnviroInsite od firmy HydroAnalysis. Jedna licence stojí 250 až 479 USD. Tento software je velmi flexibilní a dokáže zobrazovat veškerá hydrogeologická data. [4]

Práce s ním je intuitivní. Program zobrazuje data, která uživatel načte do databáze s danou strukturou. Databáze může být implementována v programu MS Access 2003 či 2007 (přípona .mdb nebo .accdb) či MS Excel. Software zobrazuje dokumentaci jednotlivých vrtů, geologické řezy, 3D vizualizaci geologie, mapy a chemické interpretační grafy (Piper, Stiff, Schoeller) umístěné na mapě nebo na samostatném listu. Podobným způsobem zobrazuje souhrnné i detailní tabulky; grafy a tabulky

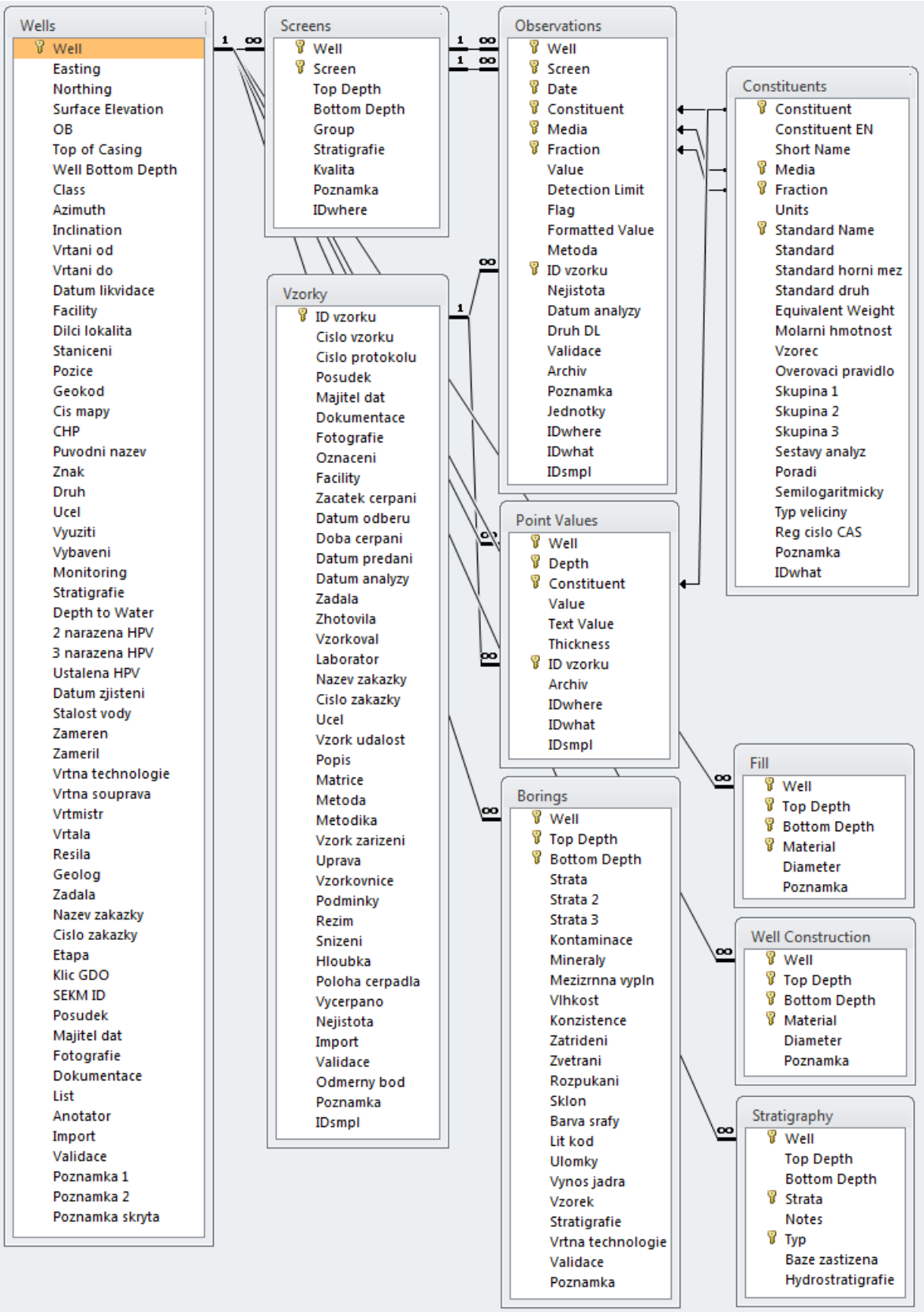
časových řad. Software interpoluje data ve 2D i 3D a je možno do něj načíst běžné formáty geografických dat (shp, dxf, dwg, rastrové obrázky). [4]

1.3.1 Rozšířený formát aplikace EnviroInsite

Jedná se o formát používaný aplikací EnviroInsite rozšířený o data pro potřeby IS MARE. Data bude tedy možno načítat a zobrazovat v EnviroInsite, ale zároveň zde budou informace navíc pro jiné formy zpracování. Soubory jsou načítány do EnviroInsite ve formátu Microsoft Access či Microsoft Excel. Jelikož se jedná o relační databázi, jsou data rozdělena do několika tabulek.

Tab. 3: Popis vybraných datových tabulek rozšířeného datového modelu EnviroInsite

Tabulka	Popis
Wells	Identifikace objektů, zejména vrtů
Screens	Vzorkovaný hloubkový interval
Vzorky	Data o měřených vzorcích
Observations	Jednotlivá měření vázaná ke vzorkovanému hloubkovému intervalu
Point Values	Jednotlivá měření vázaná ke konkrétní hloubce ve vrtu
Borings	Popis geologických vrstev
Constituents	Soupis analytů a měřených veličin
Fill	Obsyp a těsnění vrtu
Well Construction	Výstroj vrtu
Stratigraphy	Interpretované vrstvy pro geologický řez



Obr. 1: Struktura vybraných datových tabulek rozšířeného datového modelu EnviroInsite

2 Nástroje ETL (Extract, Transform, Load)

Zkratka ETL je tvořena třemi počátečními písmeny tří anglických slov – extract, transform, load. Jedná se tedy o tři části určitého procesu, který bude v následujících odstavcích popsán.

Extract neboli extrakce znamená získání dat ze zdrojových systémů. Většinou se jedná o tu nejdůležitější část ETL, jelikož to není jednorázová akce, ale činnost, která je prováděna po delší časový interval. Cílem tohoto úkonu je získání dat pro další zpracování v systému. [5]

Transform, česky transformace, je druh procesu, při němž se data získaná ze zdrojových systémů zpracovávají tak, aby odpovídala požadavkům výstupních formátů. Jedná se o celou řadu operací od konverzí, filtrování, normalizace, denormalizace, matematických operací, až po vytváření složitých struktur. V procesu transformace dochází také ke kontrole zpracovávaných dat, neboť získaná data mohou obsahovat chybné údaje. [5]

Poslední složkou ETL je load, česky naplnění. V této fázi se jedná o nahrání zpracovaných dat do cílového formátu, který je dále využíván v různých systémech a aplikacích, v tomto případě v aplikaci EnviroInsite. [5]

Nástroje ETL jsou často spojovány s výrazem Business Intelligence (BI). BI je soubor dovedností, znalostí, technologií, aplikací, kvalit, rizik, bezpečnostní otázek a postupů používaných v podnikání pro získání lepšího pochopení chování na trhu a obchodních souvislostí. Za tímto účelem se provádí sběr dat, která jsou zpracovávána. V současném vysoce konkurenčním prostředí představuje informovanost jednu z hlavních konkurenčních výhod. K těmto účelům jsou využívány právě nástroje ETL. [6]

2.1 Výběr open source nástroje ETL

Dnes je na trhu několik desítek ETL nástrojů, jejichž společným cílem je ulehčit a zefektivnit vývoj ETL procesů. Liší se v mnoha aspektech jako je funkcionalita a architektura, ale hlavně se liší licencí a cenou. Mým úkolem bylo vybrat vhodný open source nástroj ETL. Jedná se o takový software, ke kterému je k dispozici zdrojový kód, spolu s právem tento software používat, modifikovat a distribuovat.

Při konzultacích s Mgr. Kamilem Nešetřilem mi byly navrženy dva nástroje ETL. Jedná se o aplikace od společností Pentaho a Talend. V případě Pentaho je to produkt Pentaho Data Integration, známý také pod názvem Kettle. Aplikace od společnosti Talend se jmenuje Talend Open Studio for Data Integration (dostupný z webových stránek Talend [7]).

Tyto nástroje toho mají mnoho společného. Oba mají příjemné uživatelské rozhraní, širokou uživatelskou základnu a v neposlední řadě k nim existuje mnoho dokumentací, které práci s nimi ulehčují. K aplikaci Kettle existuje verze, která je primárně zaměřena na práci s prostorovými daty. Jedná se o nástroj GeoKettle. Aplikace Talend má ke stejným účelům volně dostupné rozšíření Spatial Data Integrator.

Hlavním rozdílem je, že Kettle je interpretem procedur ETL ve formátu Extensible Markup Language (XML), zatímco Talend Open Studio generuje kód v jazyce Java nebo Perl.

Nakonec byl vybrán open source nástroj Kettle od společnosti Pentaho, díky příjemnějšímu uživatelskému rozhraní a jednodušší práci se samotnou aplikací.

Kettle je šířen pod licencí Apache 2.0, která umožňuje uživateli software svobodně užívat, šířit, modifikovat a šířit upravené verze softwaru. V případě šíření upravené verze je vyžadováno přiložení kopie licence Apache.

2.2 Pentaho Data Integration (Kettle)

Největší roli při práci v aplikaci Kettle hrají tzv. **kroky**[2], pomocí kterých se sestavují transformace. Jedná se o prvky, které mají specifické funkce. Každý krok má k dispozici okno *Edit step,* kde je možno specifikovat nastavení. Nejčastěji používané kroky v transformacích této bakalářské práce a jejich funkce jsou popsány v kapitole 3.1.

[2] V následujícím textu je termín **krok** (v originále **step**) užíván vždy pro „stavební blok“ v aplikaci Kettle

Při zakládání vlastního programu jsou nejčastěji využívány dvě možnosti. První z nich je *transformation* (v následujícím textu *transformace*), ve které se pomocí kroků sestavuje samotný program pro převod dat.

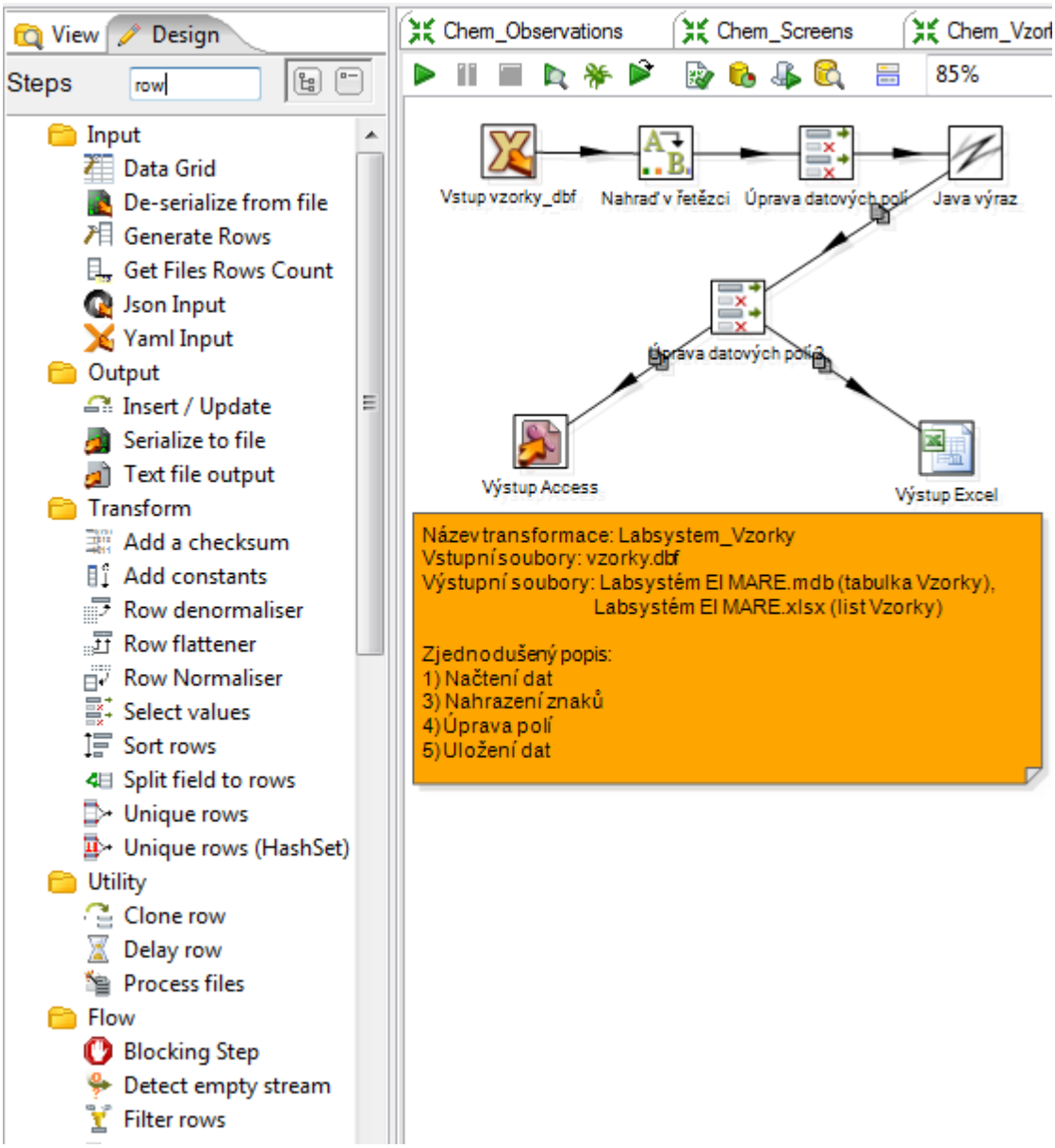
Druhou volbou je založení *job*, kde jsou převážně prvky pro podporu a správu samotných transformací, jako je odesílání e-mailů při chybě v transformaci, spouštění transformací a kontrola souborů. Některé prvky se zde nenazývají *kroky* (*steps*), ale *job entries*.

2.2.1 Transformace

Při tvorbě transformací uživatel využívá kroků, které spojuje šipkami. Ty určují směr, jakým transformace probíhá. Všechny kroky jsou v postranní nabídce aplikace a jsou rozděleny podle kategorií (Input, Output, Joins apod.). Počet kroků v nabídce přesahuje hranici 150 položek. Uživatel má tedy mnoho možností, jak s daty pracovat. K dispozici jsou i kroky, které umožňují použít v transformaci vlastní Java kód nebo příkaz Structured Query Language (SQL).

2.2.2 Job

Job je jakási obecná forma transformace, která se využívá především pro správu vytvořených transformací a požadavků s nimi souvisejících. Nabídka job entries je zde o poznání menší než nabídka kroků u Transformation. Job entries jsou opět roztříděny do několika kategorií (General, Mail, File Management apod.). Některé funkce job entries jsou velice užitečné. Například je možno spustit několik transformací současně nebo sekvenčně, kontrolovat soubory anebo si nechat zaslat e-mail s informacemi o transformacích.



Obr. 2: Nabídka kroků (levý sloupec) po zadání výrazu "row"

3 Převod dat pomocí nástroje Kettle

Všechna vstupní data jsou převáděna do požadovaného formátu pomocí výše zmíněného nástroje ETL – PDI (Kettle). Návrh převodů vznikl převážně při konzultaci s Mgr. Kamilem Nešetřilem. V případě vstupních dat z databáze řízené programem Labsystém se jedná o dvě vytvořené transformace a jeden job. V případě dat z archivu Geofond se jedná o šest transformací a jeden job.

Transformace dat z Labsystému jsou popisovány v kapitolách 3.2.1 a 3.2.2. V kapitole 3.2.3 je popsán job, sloužící pro spouštění těchto transformací.

Všechny transformace dat z archivu Geofond mají podobnou strukturu, která je popsána níže (viz kapitola 3.3). Jednotlivé transformace ale mají svá specifika, která jsou popisována v kapitolách 3.3.1 až 3.3.6. Job k těmto transformacím je popsán v kapitole 3.3.7.

Popis vytvořených transformací se zaměřuje zejména na důležité, problematické a specifické části. Ke každému vytvořenému programu je přiložen snímek obrazovky dané transformace z aplikace Kettle.

Názvy kapitol, ve kterých jsou popisovány jednotlivé programy, jsou pojmenovány dle toho, jestli byl daný program vytvořen pomocí *job* či *transformace* a dle samotného názvu programu v aplikaci Kettle.

3.1 Přehled nejčastěji používaných kroků v aplikaci Kettle

V této kapitole je v tabulce 4 uveden přehled nejčastěji využívaných kroků z aplikace Kettle, které byly použity v transformacích této bakalářské práce. V prvním sloupci tabulky 4 je jméno konkrétního kroku, ve druhém je jeho grafické značení a ve třetím je zjednodušený popis, k čemu daný krok slouží.

Tab. 4: Přehled nejčastěji používaných kroků

Název	Značení	Popis
Microsoft Access Input		Čte data ze souborů ve formátu Microsoft Access.
Microsoft Access Output		Ukládá data do tabulky databáze Microsoft Access.
Microsoft Excel Writer		Ukládá data do dokumentu ve formátu Microsoft Excel.
Sort rows		Seřadí data ve vybraném sloupci (sloupcích) vzestupně nebo sestupně.
Merge Join		Spojí řádky dvou načítaných vstupů pomocí vybraného klíče do jednoho výstupu. Vstupy musí být před spojením seřazeny podle vybraného klíče.
Select values		Výběr, úprava nebo vymazání polí. Možnost změnit typ, formát nebo délku dat.
User Defined Java Expression		Vlastní výraz napsaný v jazyce Java.
Add constant		Přidání jedné nebo více hodnot (konstant) do polí.
Filter rows		Filtrování dat řádků pomocí vybraných podmínek.
Split Fields		Rozdělení jednoho pole do více polí pomocí vybraných podmínek.
Unique rows		Kontrola unikátnosti řádků. Duplicitní řádky jsou vymazány.
Calculator		Vytvoření nového pole pomocí matematických zápisů.

3.2 Převod dat z Labsystému

V následujících kapitolách je popsán převod dat z Labsystému do cílového formátu. Za tímto účelem byly v nástroji Kettle vytvořeny dvě transformace a jeden job, který slouží pro spouštění vytvořených transformací.

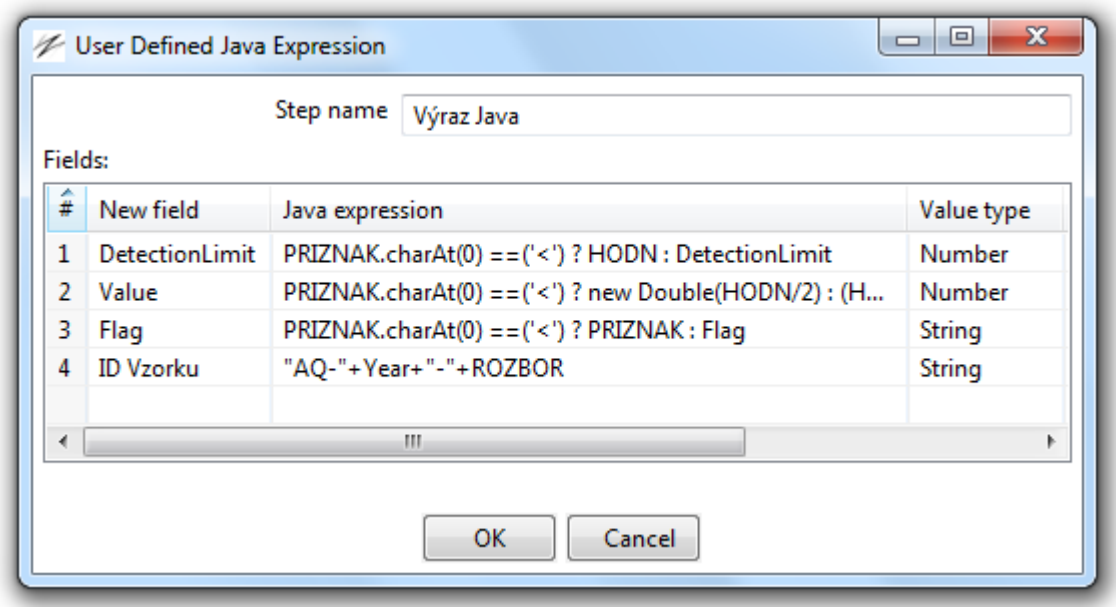
3.2.1 Transformace Labsystem_Observations

Jako vstup slouží soubory *hodnoty.dbf* a *vzorky.dbf*. Ty jsou načítány pomocí kroku *xBase Input*. Problémem těchto souborů je, že jejich data jsou uložena ve znakové sadě „kód Kamenických“, kterou Kettle neumí správně načíst, protože tuto dnes již zřídka používanou znakovou sadu nepodporuje. Místo znaků s háčky a čárkami se zobrazí jejich znakové mutace a některé znaky se nezobrazí vůbec. V nástroji Kettle sice existuje krok, který dokáže nahradit znak za znak nebo nahradit pole za pole, ale problém je právě u nezobrazovaných znaků, kde by byla nutnost všechna špatně zobrazená slova nahrazovat slovy správnými. Toto řešení je neefektivní z důvodu

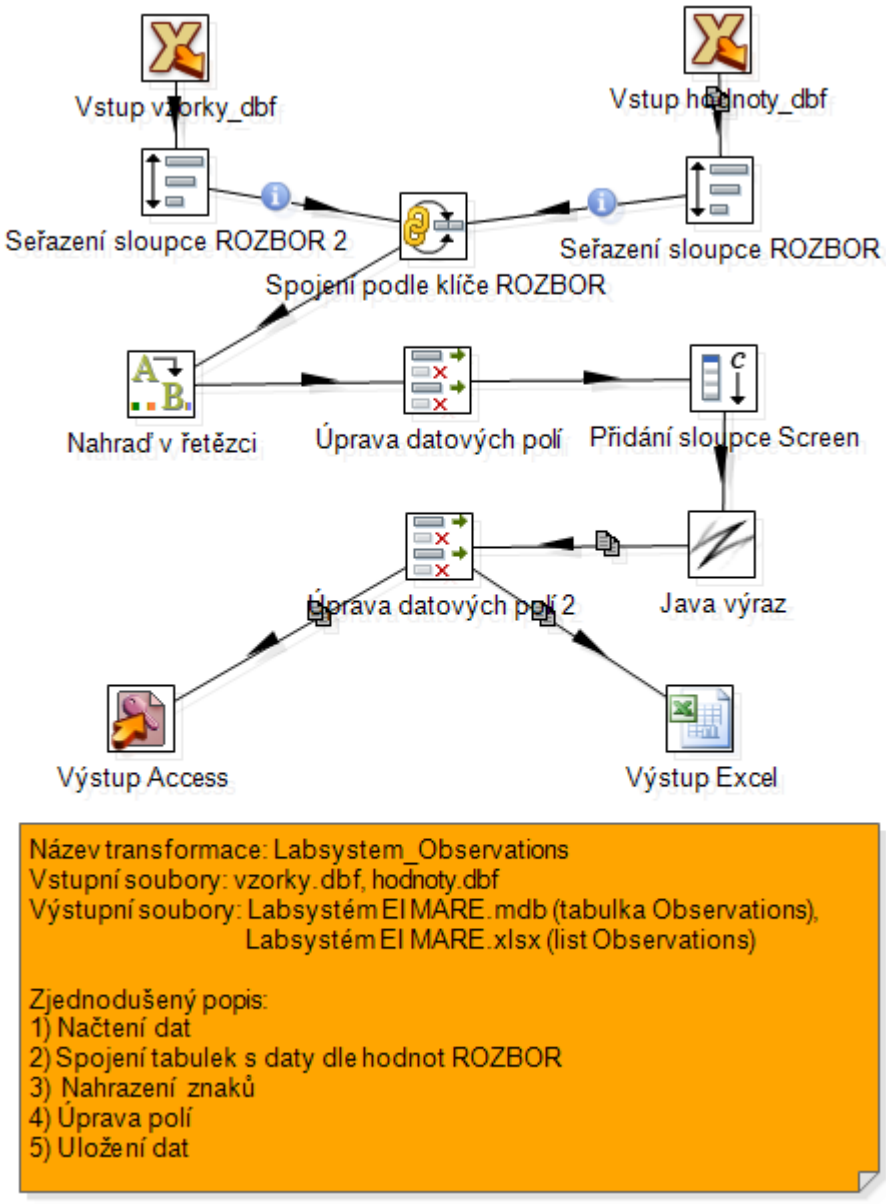
velkého počtu nahrazovaných slov a zároveň je to řešení jen pro tento jeden vstupní soubor. V případě načítání jiného vstupního souboru s jinými slovy by se některé znaky zobrazovaly znovu špatně.

Problém je vyřešen tak, že se v kroku *xBase Input* data ze souborů načítají do znakové sady IBM437, která zobrazuje většinu znaků správně, ale hlavně zobrazuje znaky všechny. Je to řešení efektivní, protože stačí v určitém kroku jen nahradit špatně zobrazené znaky za znaky správné a nemusí se nahrazovat celá slova. Řeší to i problém s případným načítáním jiných vstupních souborů.

Načítaná data z obou souborů jsou seřazena podle hodnoty ROZBOR a následně spojena pomocí kroku *Merge Join* v jednu tabulku. V následujícím kroku *Replace in String* jsou špatně zobrazené znaky načítaných dat nahrazeny za české ekvivalenty (ž, š, č, ř, ď, ť, ň, ě, í, ý a jejich velká písmena). Následuje přejmenování některých sloupců, aby názvy vyhovovaly cílovému formátu. Za pomoci vlastních kódů v kroku *User Defined Java Expression* jsou získány potřebné hodnoty pro některá pole cílového formátu. Posledním krokem je výstup v podobě souboru ve formátu Microsoft Access a Microsoft Excel.



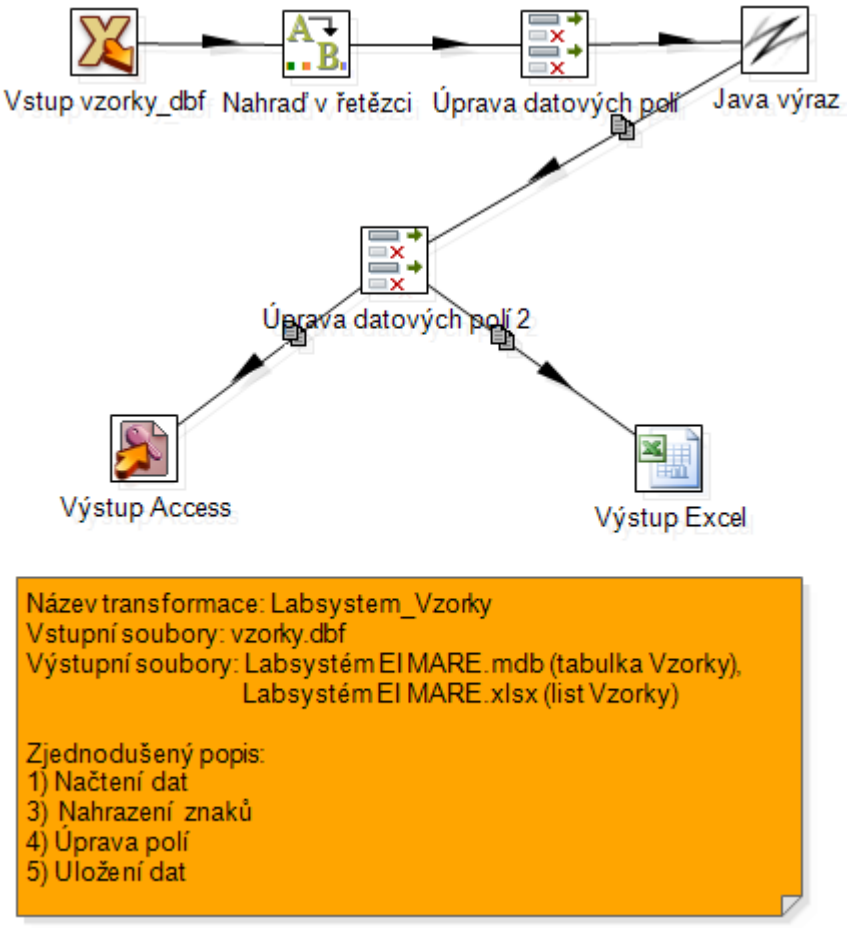
Obr. 3: Krok User Defined Java Expression použitý v transformaci Labsystem_Observations



Obr. 4: Transformace Labsystem_Observations

3.2.2 Transformace Labsystem_Vzorky

V této transformaci slouží jako vstup soubor *vzorky.dbf* a upravená data se zapisují do tabulky Vzorky. Stejně jako v předchozí transformaci je zde použit krok *Replace in String* k nahrazení špatně zobrazených znaků. Pomocí vlastního Java kódu je v kroku *User Defined Java Expression* získána hodnota *ID vzorku*. Zbylé hodnoty pro cílový formát jsou získány přejmenováním vstupních polí a úpravou jejich délky v kroku *Select values*.

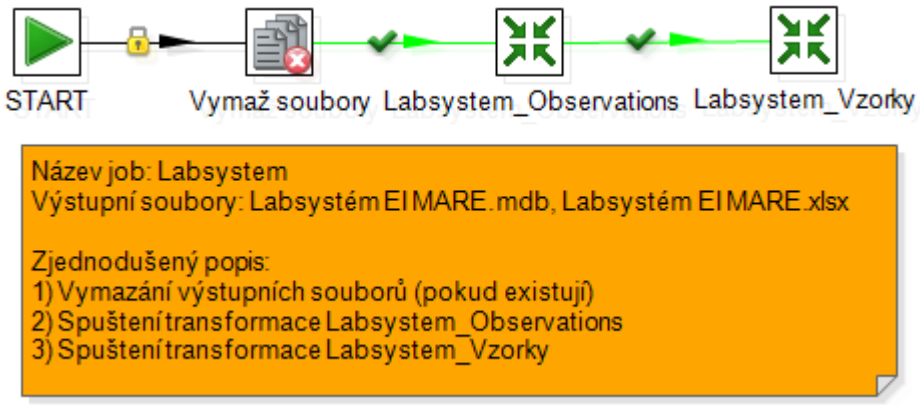


Obr. 5: Transformace Labsystem_Vzorky

3.2.3 Job Labsystem

Vytvořený job Labsystem slouží pro spuštění dvou výše popsaných transformací. Transformace je možno spustit zvlášť, ale výhodnější je spustit job *Labsystem*. V něm se vytvořené transformace postupně spustí a upravená data se zapíší do jediného souboru, protože transformace mají jako výstup nastaven stejný soubor, ale jinou cílovou tabulku.

Samotný job je tvořen pomocí kroku *Start* a pomocí job entries *Delete files* a *Transformation*. *Delete files* vymaže před samotným spuštěním transformací výstupní soubory v případě, že existují. To zabraňuje připisování stejných záznamů do stejných souborů při opakovaném spouštění vytvořeného job. Následuje postupné spuštění dvou vytvořených transformací. Výstupem jsou soubory *Labsystém EI MARE.mdb* a *Labsystém EI MARE.xlsx*.



Obr. 6: Job Labsystem

3.3 Převod dat z Geofondu

V následujících šesti transformacích jsou jako vstupní data používána data ze souborů popisovaných v kapitolách 1.2.2, 1.2.3 a 1.2.4 Data jsou v transformacích převáděna a následně ukládána do cílových tabulek, popsaných v kapitole 1.3.1. Konkrétně se jedná o tabulky *Wells*, *Screens*, *Observations*, *Borings* a *Vzorky*.

Pro každou cílovou tabulku byla vytvořena samostatná transformace, kromě tabulky *Observations*, pro kterou byly vytvořeny transformace dvě, jelikož se jedná o dvoje různá vstupní data. Každá z těchto transformací má svůj výstup v podobě souboru ve formátu Microsoft Access a Microsoft Excel. Vytvořené transformace se pak spouští a zapisují do jednoho souboru v job *Geofond*.

Všechny transformace dat z Geofondu mají podobnou strukturu a tou je:

1. načítání zdrojových souborů pomocí kroku *Microsoft Access Input*,
2. následné seřazení polí za pomoci kroku *Sort rows*,
3. spojení požadovaných dat podle kroku *Merge join*,
4. převod dat do požadované formy,
5. uložení převedených dat do souboru pomocí kroků *Microsoft Access Output* a *Microsoft Excel Output*.

Transformace Chem_Observations, Chem_Screens a Chem_Vzorky mají stejný základ v podobě načítaných souborů a následném spojení jejich tabulek. Původně byly tyto tři transformace spojeny v jednu. Jelikož ale byla tato transformace nepřehledná a data z ní se zapisovala do tří různých tabulek, byla později rozdělena na tři menší transformace.

V následujících kapitolách je popsáno všech šest vytvořených transformací a jeden job. Důraz je kladen na odlišnosti v těchto transformacích a na popis problematických míst.

3.3.1 Transformace Cerp_Observations

Jak již název napovídá, jedná se o jednu ze dvou transformací, ve kterých jsou data ukládána do tabulky *Observations*. Jako vstupy jsou v této transformaci používány soubory *lokalita.mdb* a *vrty.mdb*. V cílové tabulce *Observations* slouží jako klíč hodnoty *Well, Screen, Date, Constituent, Media, Fraction* a *ID vzorku*. Tyto hodnoty je tedy nutné mít vyplněné. Navíc k nim v této transformaci přibývají hodnoty jako *Value, Flag, Metoda* aj.

Hodnota *Well* byla získána pomocí kroku *User Defined Java Expression* jako spojení hodnot z polí *puv_nazev* (původní označení vrtu) a *posudek*. Jelikož v tabulce *Cerp* ze souboru *lokalita.mdb* nejsou původní názvy ani posudky k těmto měřením, bylo nutné spojit tento soubor pomocí kroku *Merge join* se souborem *vrty.mdb*, respektive s tabulkou *vrty*, ve které tyto hodnoty jsou.

Hodnoty *Constituent* a *Value* byly získány za pomoci kroku *Row Normaliser*, který z vybraných polí vytvoří nový sloupec na základě vazby k původním sloupcům.

Pro lepší představu je tato funkce demonstrována na příkladu v tabulce 5 a 6. Hodnoty z polí *Value* jsou následně převedeny z formátu String na formát Double.

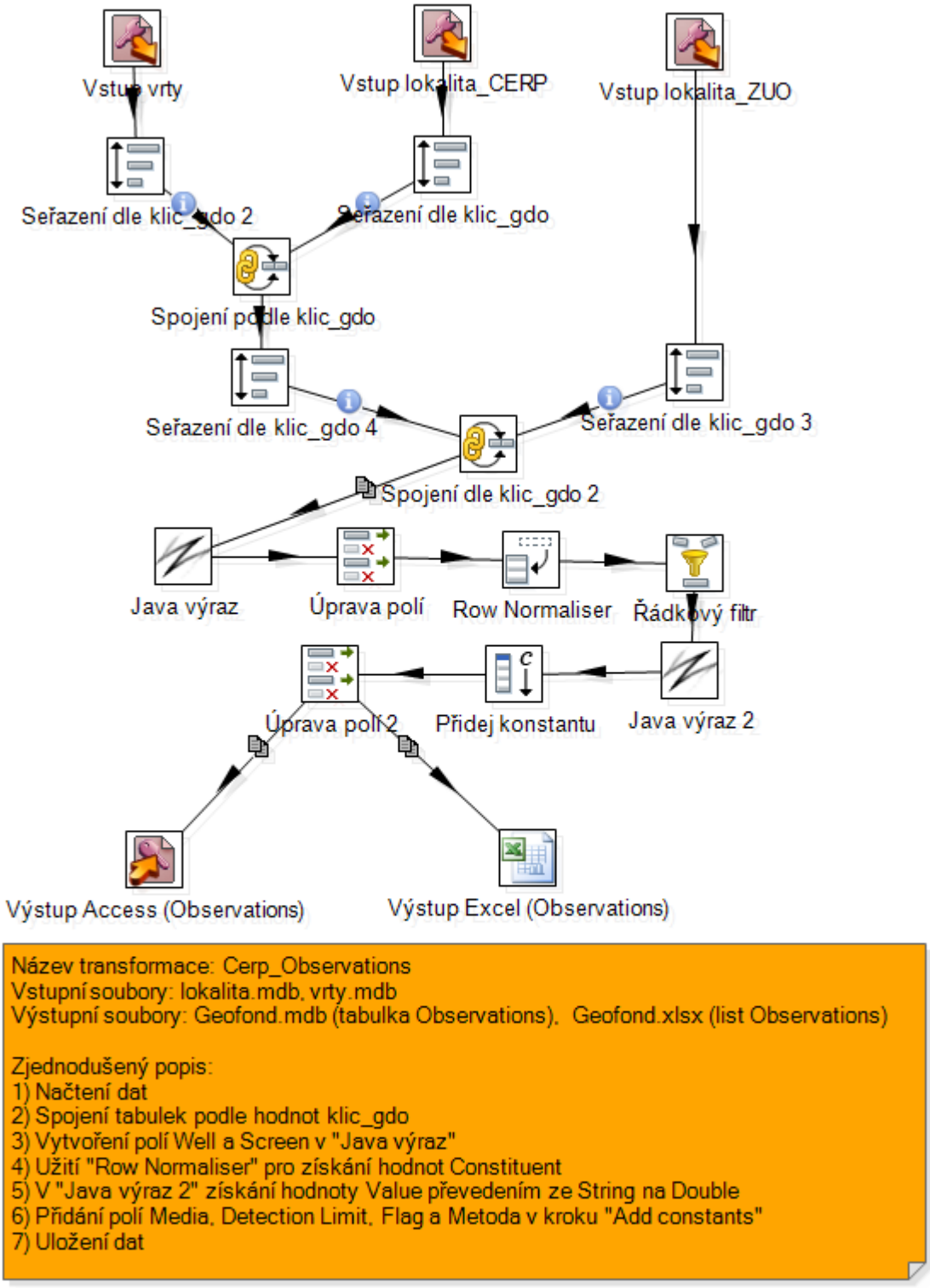
Tab. 5: Ukázka kroku Row Normaliser – vstupní tabulka

KLIC_GDO	KF1	KF2
2761	4,3	13

Tab. 6: Ukázka kroku Row Normaliser – výstupní tabulka

KLIC_GDO	Constituent	Value
2761	KF1	4,3
2761	KF2	13

ID vzorku je získáno jako spojení hodnoty *Well* a hodnoty *Datum* převedené na datový typ String ve formátu „yyyy-MM-dd HH:mm:ss“ v kroku *Select Values*. Zbylé hodnoty byly získány přejmenováním vstupních hodnot nebo použitím jednoduchých kroků.



Obr 7: Transformace Cerp_Observations

3.3.2 Transformace Chem_Observations

V tomto případě se jedná o druhou transformaci, která ukládá záznamy do tabulky *Observations*. Načítáno je pět tabulek s daty o chemických rozborech, které jsou postupně spojovány pomocí několika kroků *Merge Join* v jednu tabulku, obsahující data o všech měřeních. Kvůli chybějícím posudkům a původním názvům v těchto tabulkách je zde jako v případě transformace Cerp_Observations (viz kapitola 3.3.1) načítána tabulka *vrty* ze souboru *vrty.mdb*. Jedná se o jedinou transformaci, která se větví a následně se po několika krocích opět spojuje.

Rozpojení je z důvodu potřeby použít dvě funkce *Row Normaliser* se stejnými vstupními daty. Nejprve je tato funkce použita pro hodnoty všech měřených veličin podobně jako v případě první popisované transformace. Dále je tato funkce využita pro hodnoty metod, pomocí kterých byly zjištěny hodnoty měřených veličin. Poté, co jsou získány hodnoty měřených veličin a hodnoty jejich metod, jsou tato dvě pole přiřazena k sobě pomocí kroku *Merge Join*. Pro lepší pochopení je ukázka tohoto větvení a následného spojení popsána pomocí tabulek 7, 8, 9 a 10.

Tab. 7: Ukázka větvení – vstupní tabulka

KLIC_GDO	pH	pH_metoda	CO2	CO2_metoda
298	4,3	elektroda	13	Výpočtem

Tab. 8: Ukázka větvení – tabulka 1 po užití kroku Row Normaliser 1

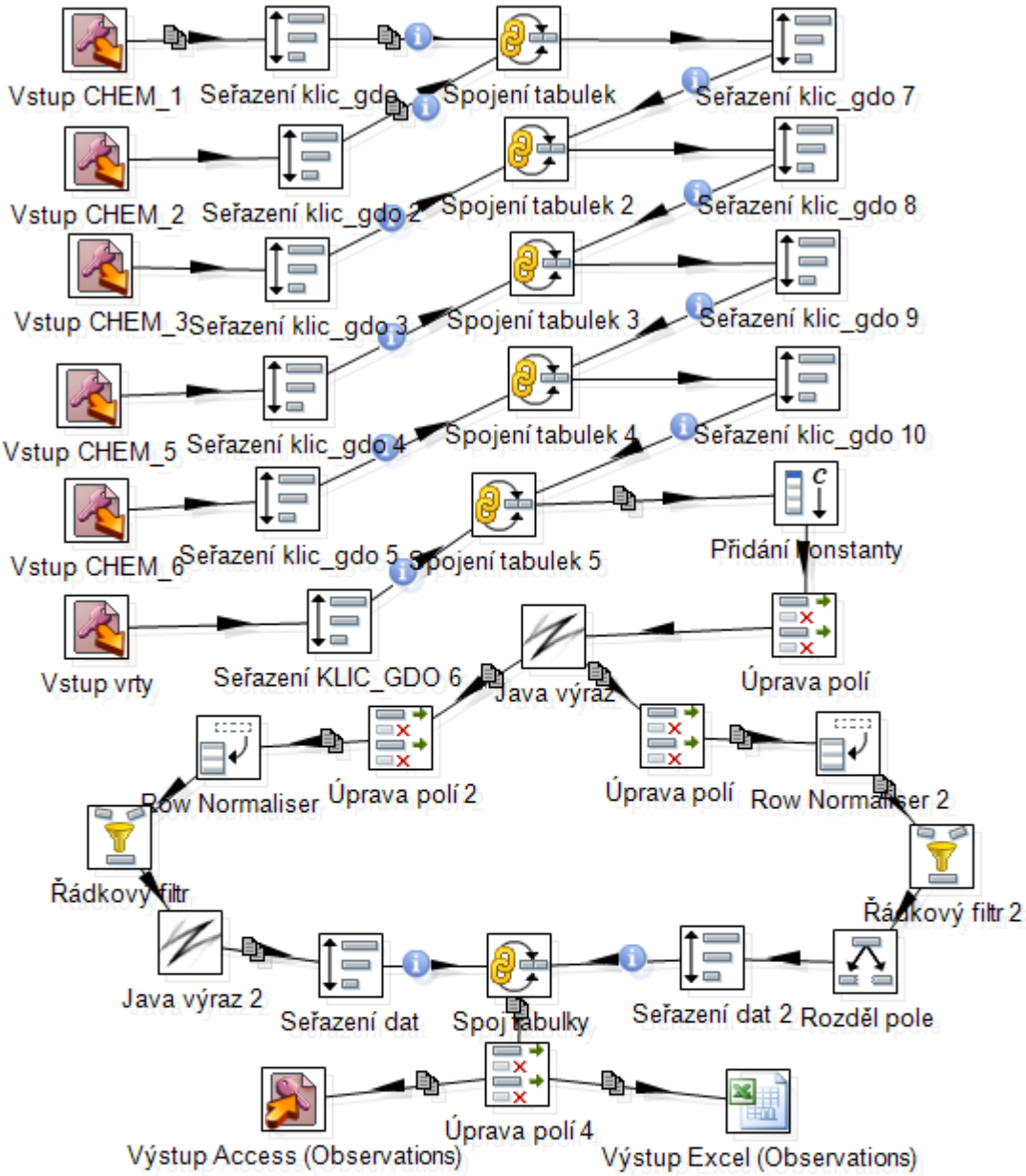
KLIC_GDO	Constituent	Value
298	pH	4,3
298	CO2	13

Tab. 9: Ukázka větvení – tabulka 2 po užití kroku Row Normaliser 2

KLIC_GDO	Constituent	Metoda
298	pH	elektroda
298	CO2	výpočtem

Tab. 10: Ukázka větvení – spojení tabulek 1 a 2 pomocí kroku Merge Join

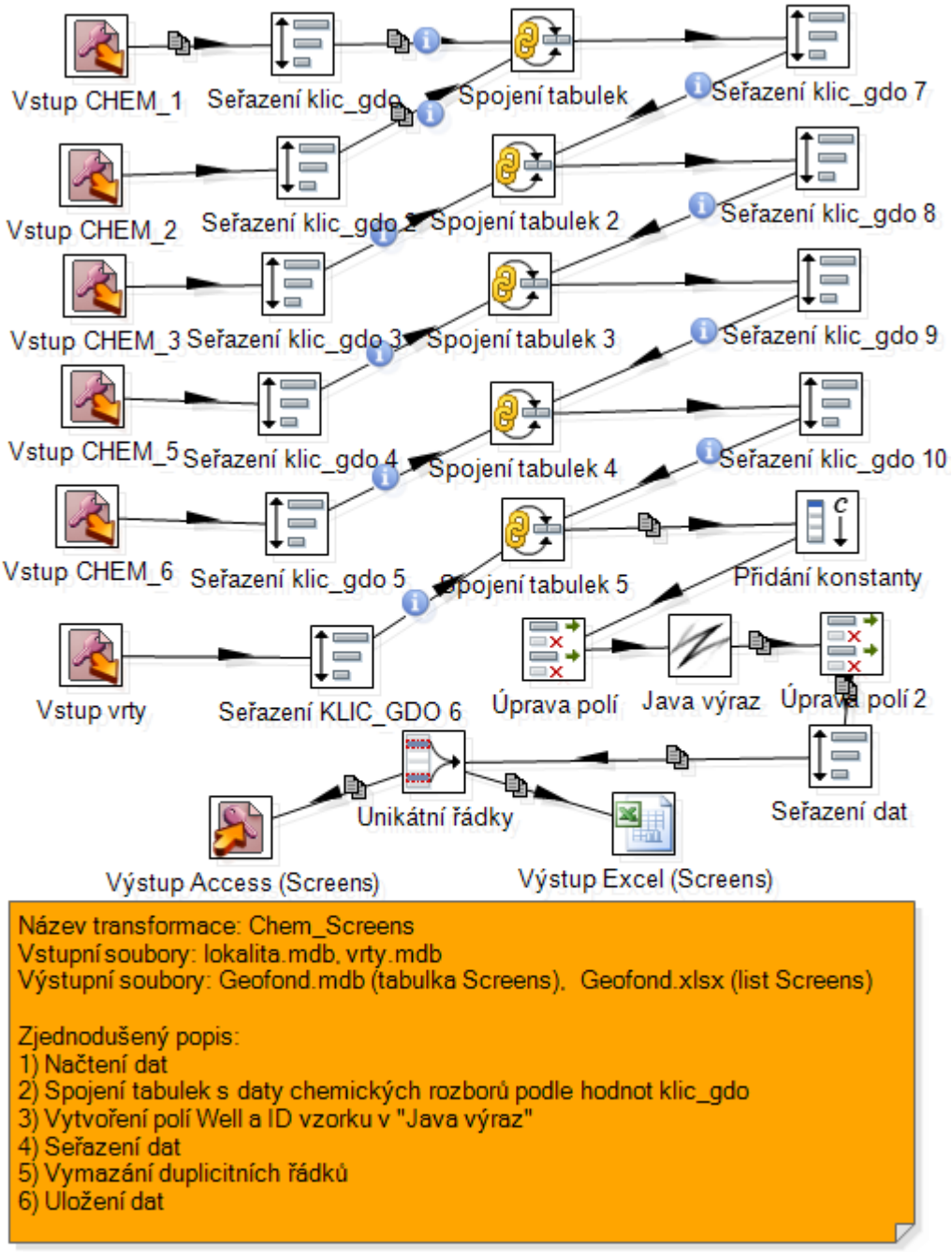
KLIC_GDO	Constituent	Value	Metoda
298	pH	4,3	elektroda
298	CO2	13	výpočtem



Obr 8: Transformace Chem_Observations

3.3.3 Transformace Chem_Screens

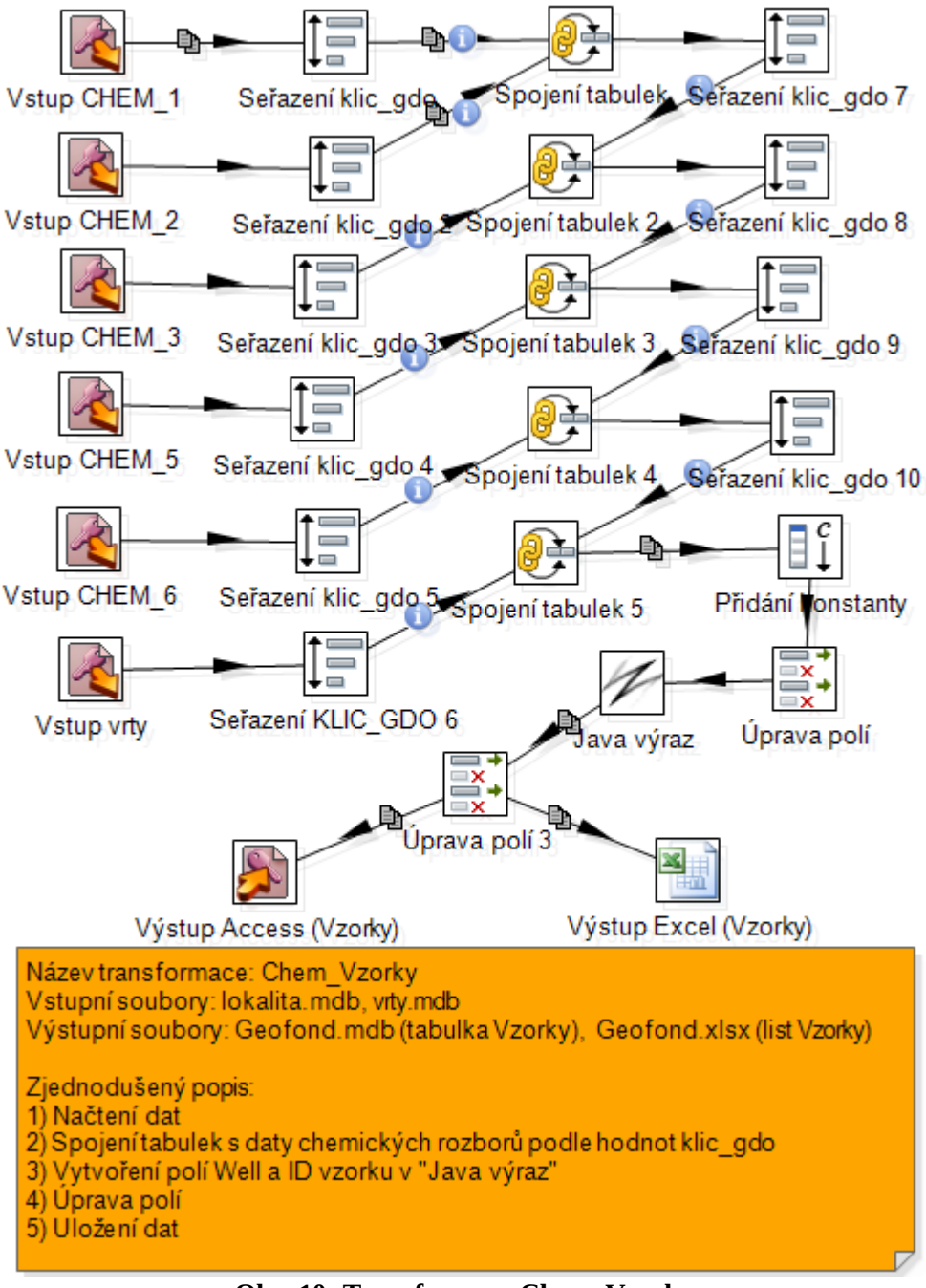
Tato transformace má stejný základ jako transformace předchozí. Liší se tím, že se zde vybírají data pro tabulku *Screens*, kde jsou klíčem hodnoty *Well* a *Screen*. Hodnoty *Well* jsou získány stejně jako v předchozích transformacích, hodnota *Screen* je přidána pomocí kroku *Add constant*. Kvůli duplicitě některých řádků je zde užit krok *Unique rows*, který duplicitní řádky maže.



Obr. 9: Transformace Chem_Screens

3.3.4 Transformace Chem_Vzorky

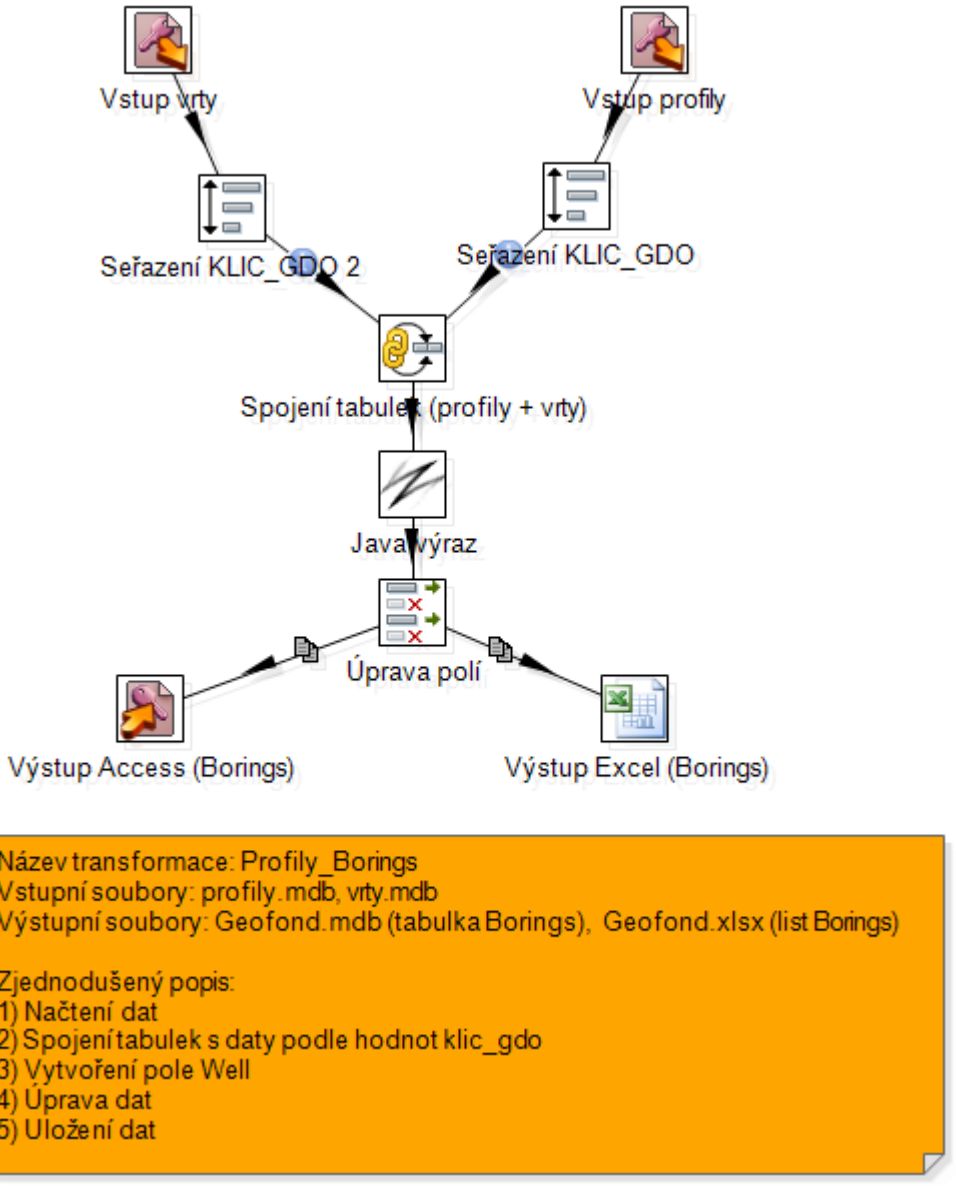
Poslední transformací se stejným základem je transformace Chem_vzorky. Jedná se o obdobu předchozích transformací s tím rozdílem, že jsou zde vybírána a následně upravována data, která se ukládají do cílové tabulky *Vzorky*.



Obr. 10: Transformace Chem_Vzorky

3.3.5 Transformace Profily_Borings

Jak již název napovídá, zdrojová data pochází ze souboru *profily.mdb* a po zpracování jsou zapsána do tabulky *Borings*. Jedná se o přepisování názvů zdrojových dat a úpravu jejich délky, aby vyhovovala koncovému formátu. Pro získání hodnoty *Well* je zde nutnost spojení tabulky *vrty* ze souboru *vrty.mdb* s tabulkou *profily*. Hodnota *Well* je získána jako spojení polí s hodnotami původních názvů a posudků.



Obr. 11: Transformace Profily_Borings

3.3.6 Transformace Vrty_Wells

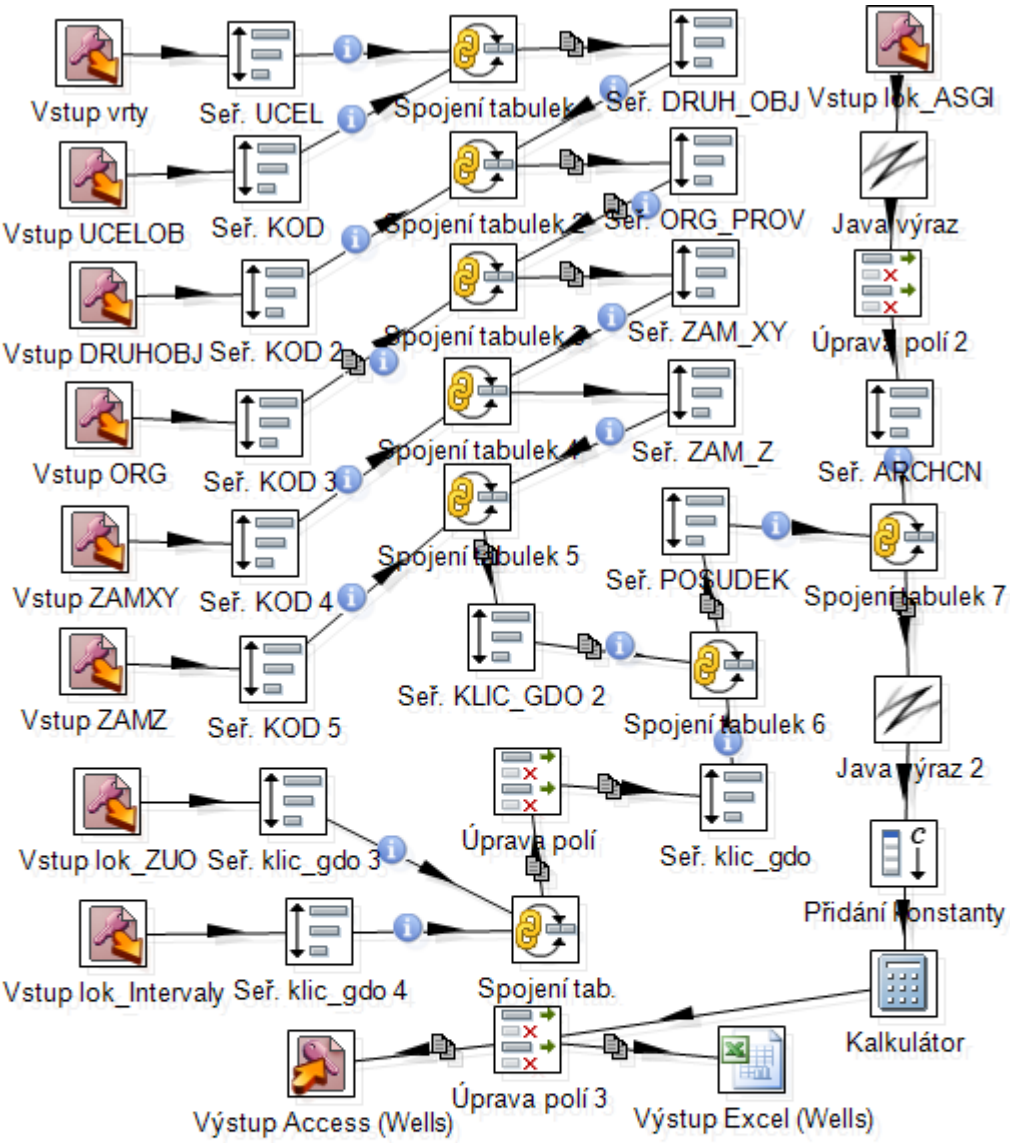
Poslední transformací, která využívá data z archivu Geofond, je transformace Vrty_Wells. Zdrojová data po upravení načítá do cílové tabulky *Wells*. Vstupními soubory jsou *vrty.mdb* a *lokalita.mdb*.

Soubor *vrty.mdb*, obsahující tabulku se záznamy o objektech, obsahuje místo některých záznamů pouze kódy, jejichž plné významy jsou uloženy v jiných tabulkách tohoto souboru (viz kapitola 1.2.2). Proto je nutné tyto soubory spojit pomocí kroku *Merge Join* a všem kódům přiřadit jejich pravé názvy.

Ze souboru *lokalita.mdb* se spojí tabulka *ZUO* s tabulkou *INTERVALY*, která obsahuje doplňující informace o objektech. Následuje spojení s daty ze souboru *vrty.mdb*.

Poslední načítanou tabulkou je tabulka *ASGI* ze souboru *lokalita.mdb*, ve které jsou uloženy záznamy o zprávách a posudcích z archivu Geofond. Jelikož jsou hodnoty těchto posudků a zpráv roztříděny do jednotlivých sloupců (např. sloupce *rok vydání*, *autoři*, *archivní číslo*), bylo nutné tyto hodnoty v každém řádku spojit pomocí vlastního Java kódu v *User Defined Java Expression* a vytvořit tak jeden řetězec, obsahující všechny informace v jednom poli. Následuje spojení s ostatními daty z předchozích kroků.

Po spojeních všech dat jsou data upravována a formátována, aby vyhovovala cílovému formátu. Zajímavým prvkem je zde užití kroku *Calculator*, ve kterém jsou hodnoty souřadnic X a Y násobeny hodnotou -1, kvůli rozdílnému souřadnicovému systému ve zdrojovém a cílovém formátu. Samozřejmostí je upravení všech nezmiňovaných polí na požadovaný název, délku a formát.



Název transformace: Vrty_Wells
Vstupní soubory: lokalita.mdb, vrty.mdb
Výstupní soubory: Geofond.mdb (tabulka Wells), Geofond.xlsx (list Wells)

Zjednodušený popis:
1) Načtení dat
2) Spojení tabulek s kódovníky
3) Spojení tabulek ZUO a Intervaly s doplňujícími daty o objektech
4) Úprava dat o posudcích z tabulky ASGI
5) Spojení všech vstupů v jednu tabulku
6) Úprava dat
7) Uložení dat

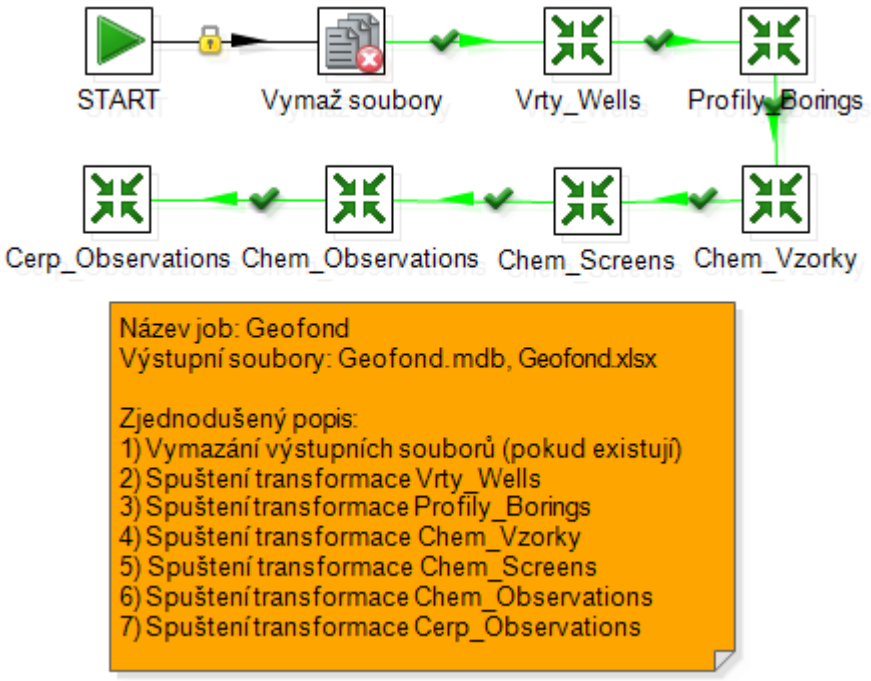
Obr. 12: Transformace Vrty_Wells

3.3.7 Job Geofond

Vytvořený job Geofond slouží pro spuštění šesti výše popsaných transformací, které využívají vstupní data právě z Geofondu. Všechny transformace je samozřejmě možno spustit zvlášť, ale výhodnější je spustit job *Geofond*, ve kterém se spustí transformace postupně.

Vytvořené transformace mají jako výstup nastaven stejný soubor, ale jinou cílovou tabulku. Výjimku tvoří transformace *Cerp_Observations* a *Chem_Observations*, které mají cílovou tabulku stejnou.

Samotný program je tvořen pomocí kroku *Start* a pomocí job entries *Delete files* a *Transformation*. Job Geofond je obdobou job Labsystem (viz kapitola 3.2.3). Výstupem jsou soubory *Geofond.mdb* a *Geofond.xlsx*.



Obr. 13: Job Geofond

3.4 Známé problémy a omezení aplikace Kettle

Následující dvě kapitoly jsou věnovány známým problémům a omezením aplikace Kettle a vytvořených transformací. Jedná se o obecně známé problémy nebo o problémy, se kterými jsem se osobně setkal.

Prvním z nich je samotné spuštění aplikace v operačním systému Windows. V případě 64bitové verze operačního systému je nutné mít nainstalováno prostředí pro spouštění aplikací Java Runtime Environment také v 64bitové verzi, jinak se aplikace Kettle nespustí.

Dalším problémem, se kterým jsem se setkal, je občasná nemožnost spuštění transformace poté, co byla při předchozím spuštění přerušena z důvodu chyby při čtení nebo zápisu do souboru. Soubor se po chybě neuzavře a následně nelze transformaci spustit. Jediným mně známým řešením je vypnutí a následné zapnutí celé aplikace.

Následující problém se týká práce v nástroji Kettle s krokem *Merge Join*, který slouží ke spojování dat. Před jeho použitím je nutné, aby uživatel data seřadil pomocí kroku *Sort rows*, což není v aplikaci uživateli při použití kroku *Merge Join* jasně řečeno. V případě, že nejsou data seřazena, krok *Merge Join* nefunguje tak, jak má.

3.5 Známé problémy a omezení vytvořených transformací

Hlavní omezení vytvořených transformací se týká vstupních dat. Všechny vytvořené transformace počítají s určitou strukturou vstupních dat a předpokládají jejich správnost. Při práci jsem narazil na některá chybná vstupní data, a tak byl výsledek samotné transformace nevyhovující cílovému formátu. Proto je při práci nutné ověřovat kromě výstupních dat i data vstupní a často tím ušetřit mnoho času hledáním neexistující chyby v transformaci.

Přestože převedená data vyhovují rozšířenému formátu aplikace EnviroInsite (dle dokumentace EI MARE), nevyhovují formátu samotné aplikace. Pro tyto potřeby by bylo třeba sjednotit názvy měřených veličin z různých vstupních zdrojů, zavést kontrolu vstupních dat (dva objekty nesmí mít stejný primární klíč) a vyřešit některé další drobné problémy. Proto má bakalářská práce velký potenciál pro případné pokračování.

4 Vlastní aplikace pro převod dat

Cílem bakalářské práce není vytvoření komplexní aplikace, která bude mít funkce podobné jako některý z nástrojů ETL. Úkolem je vytvoření jednoduché aplikace, která bude převádět vybraná data, a na které bude možnost porovnat rozdíl mezi touto aplikací a vybraným nástrojem ETL – Kettle. Pro vlastní aplikaci byl dle mých preferencí zvolen programovací jazyk Java, vývojové prostředí NetBeans a jako ukázková transformace byla vybrána transformace Profily_Borings.

4.1 Popis vlastní aplikace

V této aplikaci byla použita technologie Java Database Connectivity (JDBC), která slouží pro práci s databázemi. Poskytuje vytváření a spouštění příkazů SQL, měnících strukturu dat v databázi (CREATE, INSERT, UPDATE, DELETE) a příkazů dotazujících se nad daty (SELECT). Základem JDBC je ovladač, přes který probíhá komunikace mezi aplikací a připojovanou databází. Pro připojení ke každé databázi je třeba inicializovat vlastní ovladač. [8]

Ve vytvořené aplikaci je využíván konkrétní ovladač JDBC-ODBC bridge driver. Používání tohoto ovladače přináší řadu nevýhod. Jednou z nich je jeho nízký výkon, jelikož požadavky musí absolvovat složitou cestu. Díky tomu není tento ovladač vhodný pro internetové aplikace.

V aplikaci je využíváno metod executeUpdate a executeQuery. První jmenovaná metoda slouží pro vykonání SQL příkazů INSERT, UPDATE a DELETE. Druhá jmenovaná metoda slouží pro vykonání příkazu SELECT. Metoda executeQuery vrací objekt typu ResultSet, ve kterém jsou dotazovaná data uložena.

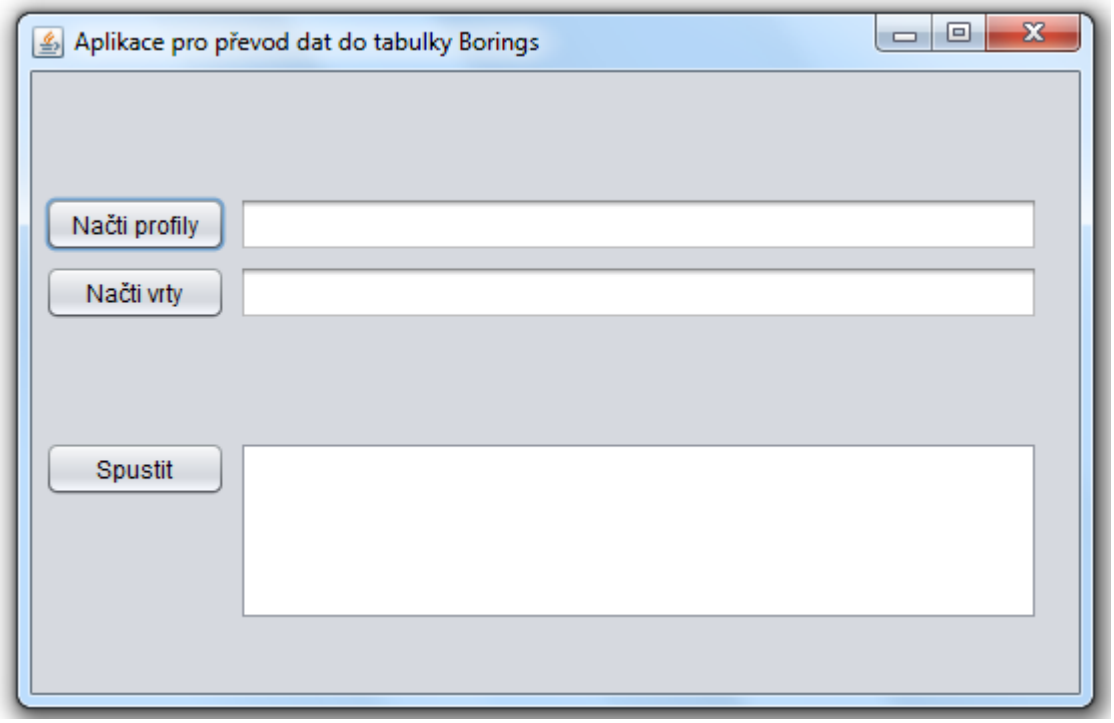
Jako první bylo vytvořeno jednoduchého uživatelské rozhraní, které má podobu panelu s tlačítky *Načti profily*, *Načti vrty* a *Spustit*. Každé tlačítko reaguje na vyvolanou akci naprogramovanou metodou. Tlačítka *Načti profily* a *Načti vrty* spouští metodu, která vyvolá okno pro výběr daného souboru. Pro snazší nalezení souborů umožňují metody uživateli vybrat soubory pouze v požadovaném formátu *mdb*.

Tlačítko *Spustit* spouští samotný převod. Předtím, než se převod spustí, je provedena kontrola, zdali jsou načteny správné soubory. V případě špatně vybraných vstupních souborů je uživatel upozorněn a převod se nespustí. V opačném případě je

vyvoláno okno, kde uživatel pojmenuje výstupní soubor a vybere cestu k jeho uložení. Poté je navázáno spojení s databázemi (se dvěma vstupními a jedním výstupním souborem) pomocí JDBC-ODBC ovladače. Následuje série SQL příkazů pro vytváření, přejmenovávání a vkládání dat mezi jednotlivými tabulkami načítaných souborů.

Hodnoty ve sloupcích *PUV_NAZEV* a *POSUDEK* ze vstupních souborů jsou z neznámých důvodů uloženy ve špatném formátu, kde textový řetězec obsahuje na konci několik přebytečných bílých znaků. Jelikož se hodnoty z těchto sloupců spojují, je nutné tyto znaky vymazat. K tomuto účelu je použita metoda *trim(),* která odstraňuje bílé znaky v řetězcích.

Po úpravě všech dat tak, aby vyhovovala cílovému formátu, jsou tato data uložena do souboru, jehož jméno a místo uložení si uživatel na začátku zvolil.



Obr. 14: Rozhraní vlastní aplikace pro převod dat

4.2 Známé problémy a omezení vlastní aplikace

Jelikož se jedná o jednoduchou aplikaci, která není primárně určena k dalšímu využití (slouží výhradně pro porovnání s prací v aplikaci Kettle), obsahuje řadu omezení, se kterými se může uživatel setkat.

Podobně jako v transformacích v aplikaci Kettle se zde počítá se správnými vstupními soubory. Ve vytvořené aplikaci volí vstupní soubory sám uživatel, a tak zde vzniká riziko špatně zvolených vstupních dat. Aplikace je opatřena kontrolou vstupních souborů podle jména těchto souborů. V případě, že uživatel vybere jiné soubory než *profily.mdb* a *vrty.mdb*, aplikace ho upozorní a samotná transformace neproběhne. Aplikace tedy nekontroluje samotná data v souboru, a tak v případě, že by uživatel přejmenoval nějaký soubor na *profily.mdb*, respektive *vrty.mdb*, a ten následně načítal, tak by aplikace skončila chybou.

Největší omezení vlastní aplikace spočívá v nastavení ODBC v operačním systému Windows. ODBC je rozhraní sloužící jako přístup k databázovým systémům. Omezení aplikací na různých architekturách operačních systémů je rozsáhlá problematika, se kterou jsem se při tvorbě vlastní aplikace potýkal.

Problémem je užívání vlastní aplikace na 64bitovém operačním systému Windows s kancelářským balíkem Microsoft Office 32bitové verze. Aplikace pak užívá 64bitového ovladače ODBC ke spojení se s 32bitovou databází MS Access, což vyústí v chybu v programu. V tomto případě je nutné nastavit v operačním systému přístup k databázi přes 32bitový ovladač ODBC, který se spustí napsáním *C:\Windows\SysWOW64\odbcad32.exe* do konzole v nabídce *Start*. Dokumentace k ODBC je dostupná na webu [9].

5 Zhodnocení práce s aplikací Kettle

Před sestavováním samotných transformací v aplikaci Kettle jsem se musel nejprve naučit jejím základům. Bylo potřeba se naučit používat základní kroky, které jsou popsány v dokumentaci, nebo se je naučit z ukázkových transformací, které jsou v aplikaci Kettle k dispozici. Poté jsem z těchto nejjednodušších kroků zkoušel sestavit jednoduché transformace a zjistit, jakým způsobem v aplikaci Kettle transformace probíhají.

Práce na samotných transformacích v bakalářské práci probíhala od těch nejjednodušších částí, jako je použití kroků pro vstup, výstup a spojování tabulek. Jako první jsem do transformací přidal právě tyto kroky a následně jsem pomocí ostatních, komplikovanějších kroků, upravoval data do podoby cílového formátu.

Velmi časté bylo také předělávání transformací z důvodu stále se měnícího rozšířeného formátu EnviroInsite. Mnohé části transformací byly postupně řešeny více způsoby. Například násobení hodnoty konstantou bylo nejdříve napsáno pomocí Java kódu v kroku *User Defined Java Expression*, ale po čase byla tato funkce z důvodu přehlednosti nahrazena kroky *Add Constant* a *Calculator*, které plní stejnou funkci.

Po vytvoření první funkční transformace byla práce v aplikaci Kettle podstatně jednodušší, protože základní struktura všech transformací si je velice podobná. V průběhu práce na ostatních transformacích se bylo potřeba naučit i složitějším krokům, které ale výrazně ulehčovaly práci, a tak nebyl později problém sestavit i komplexnější transformace.

Velkou podporou při tvorbě transformací mi bylo, kromě zmiňované dokumentace a ukázkových příkladů, internetové fórum nástroje Kettle (dostupné na webových stránkách PDI [10]), kde mi byly zodpovězeny všechny dotazy ohledně použití různých kroků či poskytnuty rady k nefunkčním transformacím.

6 Zhodnocení práce na vlastní aplikaci

Při programování vlastní aplikace jsem narážel na jistá úskalí spíše než při práci s aplikací Kettle a to i přesto, že mi byly vývojové prostředí NetBeans a programovací jazyk před začátkem práce známy.

Jedním z problémů při tvorbě aplikace bylo, že Java nepodporuje vytváření *mdb* souborů. Tuto možnost podporuje například knihovna Jackcess (dostupná ke stažení z [11]), ale nebyla použita z toho důvodu, že neposkytuje JDBC ovladač a také z důvodu nedůvěry k cizím, neověřeným zdrojům. Tento problém je vyřešen tak, že se jeden z vybraných vstupních souborů zkopíruje, přejmenuje, vymaže se jeho obsah a zapisuje se do něho, jako do souboru výstupního.

Po vyřešení všech problémů byla vlastní aplikace pro převod dat úspěšně dokončena, a poskytuje tak možnost srovnání s prací na převodech dat v nástroji Kettle.

Závěr

Jedním z úkolů této bakalářské práce bylo prozkoumat open source nástroje ETL a následně z nich vybrat vhodný nástroj, pomocí kterého se převedou obdržená vstupní data do určeného cílového formátu. Po prozkoumání problematiky nástrojů ETL jsem vybral aplikaci Pentaho Data Integration (Kettle) od společnosti Pentaho.

Po seznámení se s formáty vstupních dat z Labsystému a z archivu Geofond jsem se seznámil s rozšířeným formátem aplikace EnviroInsite i se samotnou aplikací. Pochopení struktury těchto dat bylo nutnou podmínkou k tomu, abych mohl data pomocí aplikace Kettle následně převést do požadované podoby.

Převážná část mé bakalářské práce byla spojena s prací v aplikaci Kettle, kde jsem se musel nejprve naučit základům této aplikace a až poté začít vytvářet samotné transformace. S přibývajícími zkušenostmi s prací v aplikaci Kettle jsem jednotlivé transformace neustále upravoval, aby byly co nejvíce efektivní a vyhovovaly potřebám neustále se měnícího rozšířeného formátu aplikace EnviroInsite.

Výsledky těchto transformací, resp. data výstupních souborů, byly zkontrolovány a předány konzultantovi, který ověřil, že výstupní soubory odpovídají specifikaci rozšířeného datového modelu aplikace EnviroInsite. Výstupní soubory obou jobs (transformací) jsou k dispozici na přiloženém CD ve složce *Vystupni soubory*. Jelikož jsem využil jen zlomek toho, co Kettle nabízí, tak jsem si jistý, že by některé části transformací šly vyřešit jinak a lépe.

Dalším úkolem bylo vytvoření vlastní aplikace pro převod dat. Dle mých zkušeností s programováním jsem si vybral programovací jazyk Java a vývojové prostředí NetBeans. Výsledkem je aplikace, která převádí stejná data jako jedna z transformací vytvořených v aplikaci Kettle. Program není zdaleka dokonalý a v případě dalšího používání by bylo nutné některé jeho části zdokonalit, ale pro účely této bakalářské práce je plně dostačující. Výstupem vlastní aplikace je soubor, který obsahuje stejná data jako výstupní soubor této transformace z aplikace Kettle.

Hlavním úkolem a výsledkem mé bakalářské práce bylo porovnání přístupu k programování vlastní aplikace pro převod dat a k práci na transformacích ve zvoleném nástroji ETL – Kettle. Paradoxně se mi lépe na transformacích pracovalo v aplikaci Kettle, kterou jsem předtím neznal, než v jazyce Java, prostředí pro mě známém. Při práci na transformacích v aplikaci Kettle jsem se nesetkal s problémem, který by nešel

efektivně vyřešit, zatímco při práci na vlastní aplikaci jsem narazil na několik problémů, které nebyly vyřešeny optimálně.

Z mé osobní zkušenosti je při těchto typech převodů dat vhodnější používat aplikaci Kettle, než psát vlastní aplikaci. Na rozdíl od psaní aplikace pro určitý typ převodu, kterou nelze jinak než úpravou kódu změnit, jsou transformace v aplikaci Kettle lehce modifikovatelné i pro člověka, který má s programováním malé či žádné zkušenosti. Navíc je možnost v aplikaci Kettle využít vlastních SQL příkazů, vlastního Java kódu nebo spouštět externí aplikace.

Další důvod, proč je Kettle vhodnější nástroj pro transformaci dat, je jeho přehlednost a snadná detekce chyb. Ve vlastní aplikaci pro převod dat nemusí být vždy ošetřeny všechny podmínky, a tak se může stát, že program neproběhne nebo se přeruší a uživatel neví proč. V aplikaci Kettle se chybová hlášení zobrazí přímo v kroku, ve kterém nastala chyba, takže má běžný uživatel možnost chybu opravit. Je to tedy vývojové prostředí pro transformace s možností ladění přímo v rozhraní samotné transformace.

Výhoda vlastní aplikace je v její jednoduchosti, kdy nemá uživatel oproti nástroji Kettle prakticky žádnou možnost, jak převod upravit, tudíž zde není téměř žádný prostor pro chybu ze strany uživatele.

Na základě této bakalářské práce se řešitelé projektu MARE rozhodli používat v rámci projektu aplikaci Pentaho Data Integration – Kettle.

Aplikace Kettle je vhodnější na transformace dat, zatímco přednost vlastní aplikace spočívá v její jednoduchosti. Proto by bylo vhodné na práci navázat a zkombinovat oba užité přístupy k vytvoření nástroje pro další potřeby projektu MARE.

Seznam použité literatury

[1] Profil laboratoře: Aquatest. Aquatest [online]. © 2009 [cit. 2013-04-11]. Dostupné z: http://www.aquatest.cz/cz/portfolio-sluzeb/laboratorni-sluzby/profil-laboratore/

[2] Státní geologická služba. Česká geologická služba [online]. 2001 [cit. 2013-04-12]. Dostupné z: http://www.geology.cz/extranet/sgs

[3] Whitespace characters. *Wikipedia, the free encyclopedia* [online]. 2001, akt. 28.3. 2013 [cit. 2013-05-04]. Dostupné z: http://en.wikipedia.org/wiki/Whitespace_character

[4] Jan Šembera, Miroslav Černík, Lenka Lacinová, Kamil Nešetřil, Jaroslav Nosek, Štěpánka Klímková, Vratislav Žabka, Dana Rosická, Eva Kakosová. *Roční výzkumná zpráva o výsledcích řešení IV. tématického okruhu projektu FR-TI1/456, Vývoj a zavedení nástrojů aditivně modulujících proces bioremediace půdy a vody*. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií. Ústav nových technologií a aplikované informatiky (NTI). 74 stran. [předáno Kamilem Nešetřilem]

[5] Co se skrývá pod zkratkou ETL?. *Ekonomické a informační systémy v praxi* [online]. © 2001 - 2013 [cit. 2013-01-10]. Dostupné z: http://www.systemonline.cz/clanky/co-se-skryva-pod-zkratkou-etl.htm

[6] Průmyslové informační systémy. *Agent Technology Center* [online]. 2010 [cit. 2013-04-12]. Dostupné z: http://exile.felk.cvut.cz/wiki/lib/exe/fetch.php?id=teaching%3Apis&cache=cache&media=teaching:pis:predn:a0m33pis-12.pdf

[7] Talend. Talend Inc. Spatial module for Talend Open Studio. [online]. [cit. 2013-03-17]. Dostupné z: www.talendforge.org/wiki/doku.php?id=sdi:MainPage

[8] Referáty: jdbc. *Čečák* [online]. 2009 [cit. 2013-05-05]. Dostupné z: http://www.cecak.cz/fel/dba/referaty/jdbc

[9] Microsoft Open Database Connectivity. *Windows Desktop Development* [online]. © 2013 [cit. 2013-05-03]. Dostupné z: http://msdn.microsoft.com/en-us/library/windows/desktop/ms710252%28v=vs.85%29.aspx

[10] Pentaho Data Integration [Kettle]. *Pentaho Community Forums* [online]. © 2005 – 2011 [cit. 2013-04-27]. Dostupné z: http://forums.pentaho.com/forumdisplay.php?135-Pentaho-Data-Integration-Kettle

[11] Java Library for MS Access. *OpenHMS Software* [online]. © 2007 [cit. 2013-03-23]. Dostupné z: http://jackcess.sourceforge.net/

[12] Kettle Project: Pentaho Data Integration. *Business analytics and business intelligence leaders - Pentaho* [online]. © 2005 - 2013 [cit. 2013-05-10]. Dostupné z: http://kettle.pentaho.com/

Příloha A – Uživatelský návod pro práci s vytvořenými transformacemi v aplikaci Kettle

Na přiloženém CD je v adresáři *Kettle* soubor *pdi-ce-4.4.0-stable.zip*, který rozbalíme kamkoliv na disk. Obsahuje aplikaci Kettle staženou z webových stránek PDI [12]. Následně zkopírujeme složku *transformace_jobs* taktéž kamkoliv na disk.

Aplikace Kettle se spustí souborem *Spoon.bat* ze složky, do které jsme aplikaci rozbalili. Jednotlivé transformace či jobs otevřeme kliknutím na tlačítko *File – Open* a poté vybereme soubory s transformacemi či jobs ze zkopírovaného adresáře *transformace_jobs*. V tomto adresáři jsou dva podadresáře – *Geofond* a *Labsystem*. Soubory s transformacemi (jobs) se nachází v jednotlivých podadresářích nazvaných podle jména transformací (jobs) popisovaných v bakalářské práci.

Cesty vstupních souborů jsou ve vytvořených transformacích (jobs) nastaveny pomocí regulárních výrazů na relativní, takže se vstupní soubory ze zkopírovaného adresáře *transformace_jobs* samy načtou.

Jestliže chceme spustit transformace dat z Geofondu, tak pomocí *File – Open* najdeme cestu k souboru *Geofond.kjb* (adresář *transformace_jobs/Geofond/Job*). Tím otevřeme job, který po stisknutí klávesy F9 nebo po kliknutí na zelený trojúhelník  v horním panelu hlavního okna programu spustí postupně všechny vytvořené transformace. Následně klikneme na *Launch*. Výstupní soubory ve formátu *mdb* a *xlsx* se zapíší do adresáře *transformace_jobs /Geofond/Vystup*.

Pokud chceme spustit transformace dat z Labsystému, tak otevřeme soubor *Labsystem.kjb*. Poté klikneme na zelený trojúhelník v horním panelu hlavního okna programu nebo zmáčkneme klávesu F9. Výstupní soubory se nachází v adresáři *transformace_jobs/Labsystem/Vystup*.

Výstupem job je soubor, který obsahuje data v jednotlivých tabulkách dle vytvořených transformací. Samozřejmě je možné spustit jednotlivé transformace zvlášť. Po spuštění transformace bude výstupní soubor obsahovat pouze datovou tabulku dané transformace. Opakovaným spouštěním stejné transformace se data připisují do datové tabulky, která již tato data obsahuje. Proto je výhodnější používat vytvořené jobs, které výstupní soubory (pokud existují) před spuštěním transformací smažou.

Příloha B – Uživatelský návod pro práci s vlastní aplikací

Vytvořená aplikace v rámci této bakalářské práce se nachází v adresáři *Vlastni aplikace* a spustí se souborem *Prevod.jar*. Pro správný chod této aplikace je třeba mít nainstalovaný MS Access a správně nastavený zdroj dat ve správci zdrojů dat ODBC v operačním systému Windows (viz kapitola 4.2).

Po spuštění aplikace klikneme na tlačítko *Vyber profily*. Následně vybereme cestu k souboru *profily.mdb* uloženém v adresáři *Vlastni aplikace/Vstupni data*. Po kliknutí na tlačítko *Vyber vrty*, vybereme soubor *vrty.mdb* z téhož adresáře. Po kliknutí na tlačítko *Spustit* napíšeme do formuláře název výstupního souboru a vybereme cestu, kam ho chceme uložit. Poté proběhne samotná transformace.