**Zlepšovací návrh - alternativní metoda porovnávání dat – zákl. dokumentace**

1. **Důvod vytvoření metody křížového porovnávání a její implementace**

Aktuální stav: Běžně používané metody a nástroje pro porovnávání tabulek (např. univerzální a jinak skvělý program Beyond Compare) předpokládají, že porovnávané tabulky mají přibližně podobnou strukturu – stejné pořadí sloupečků a řádků. Pokud tabulky obsahují záhlaví, programy předpokládají, že tato záhlaví mají stejný počet řádků.

Předkládaný zlepšovací návrh sestává ze dvou částí: Z metody, která řeší tyto nedostatky a z její implementace ve formě spustitelného programu pro operační systémy MS Windows.

Projekt HR-Core: Jedním z mých dosavadních úkolů v rámci projektu HR-Core (jehož týmu jsem členem) bylo poskytnout HR uživatelům nástroj pro ověřování míry úspěšnosti migrace dat mezi velmi odlišnými datovými platformami (MSSQL vs SAP). Nástroj měl minimalizovat potřebu „ručního“ porovnávání jednotlivých hodnot samotnými uživateli.

Pokud by struktura porovnávaných tabulek byla podobná, bylo by možné k realizaci tohoto úkolu použít zmíněný a standardně dostupný nástroj Beyond Compare. Ukázalo se ale, že tento program není pro zmíněný úkol vhodný, protože vstupní a výstupní tabulky ověřované migrace mají odlišné pořadí sloupců i řádků, jiné názvy sloupců v jejich hlavičkách a navíc rozílný počet řádků v hlavičkách. Bez náležité a časově náročné úpravy tedy tyto soubory zřejmě nelze programem Beyond Compare efektivně porovnat.

Prototyp: Z uvedeného důvodu byla tedy nejprve definována jednoduchá metoda (viz níže), která pomáhá různé důsledky zmíněných problémů buďto obejít, nebo pokud možno, zcela vyřešit. Následně byl vytvořen prototyp, který tuto metodu implementuje v jazyku VBA (jazyk pro automatizaci funkcí obsažených především v aplikacích kancelářského balíku MS Office). Tato implementace fungovala správně, ale z důvodu použité běhové platormy (interpretovaný VBA) pracovala *velmi* pomalu. Pro soubory obsahující několik desítek tisíc řádků se ukázala být, především z hlediska požadovaného uživatelského pohodlí, zcela nepoužitelná.

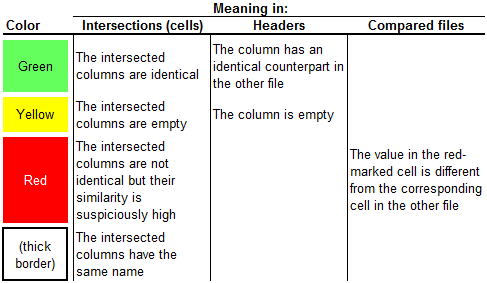
Konečná implementace: Z důvodu diskutované velice nízké efektivity VBA implementace bylo tedy rozhodnuto o vytvoření univerzálně použitelného nástroje, který by mohl pracovat i s velmi velkými objemy dat. Tento nástroj byl vytvořen v jazyku C++ na platformě MS Windows s využitím knihovny MFC (Microsoft foundation Class). Toto řešení umožňuje programu komunikovat a „úkolovat“ jím spuštěnou novou instanci programu MS Excel.

Výsledek: Aktuální verze programu **XCompare** 1.1.8.1 (15.10.2017) je dostatečně optimalizována jak na úrovni užitých algoritmů (použití pokročilých datových kontejnerů) tak i programového kódu (kde to je účelné, je běh programu větven do nezávislých vláken za účelem lepšího využití vlastností moderních CPU) pro efektivní používání i s velmi velkými (několik set tisíc rádků) tabulkami.

**XCompare** (Cross compare) plní svůj úkol významně rychleji, než univerzální nástroj Beyond Compare.

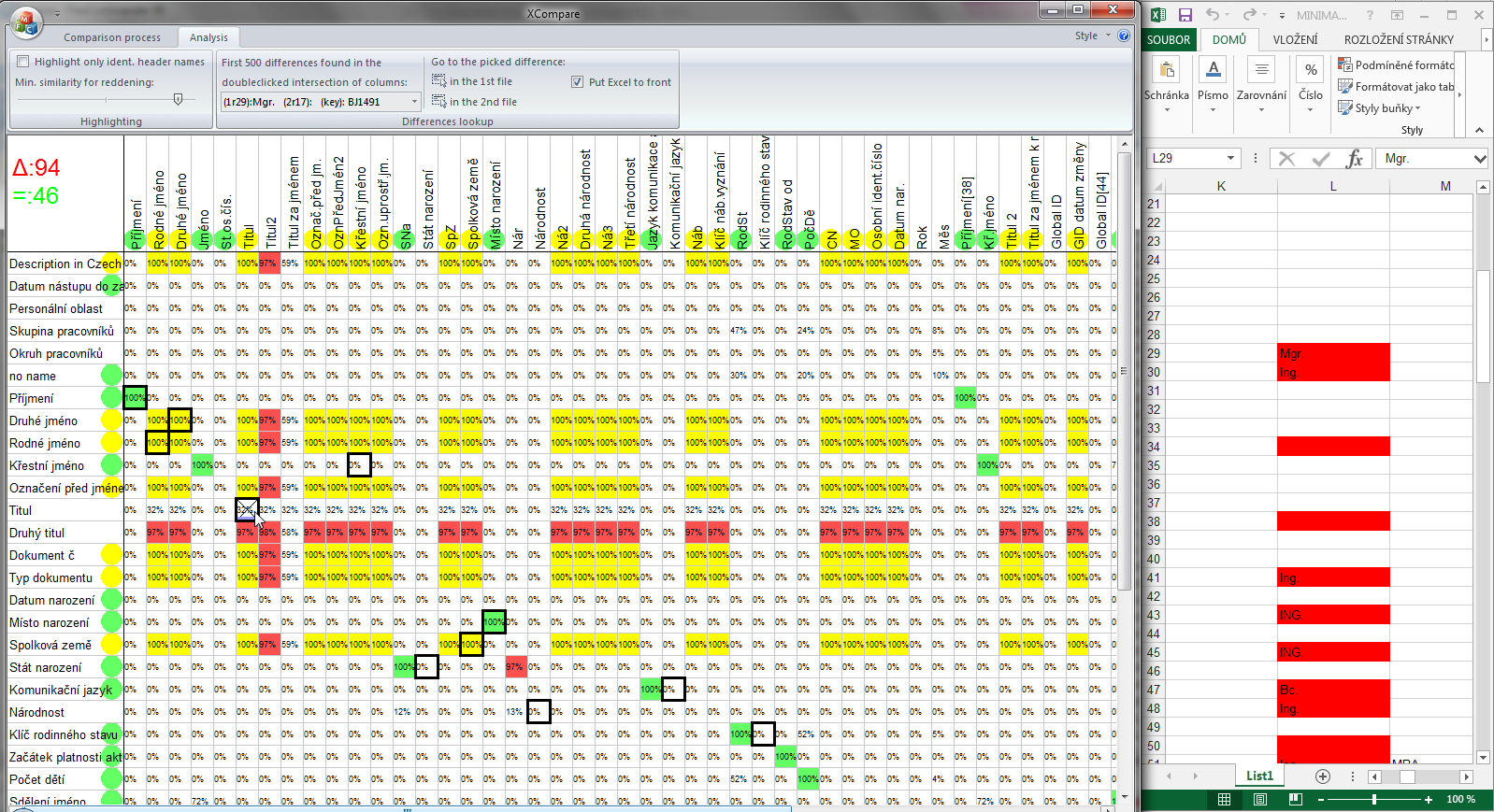
**2) Předpokládané scénáře použití:**

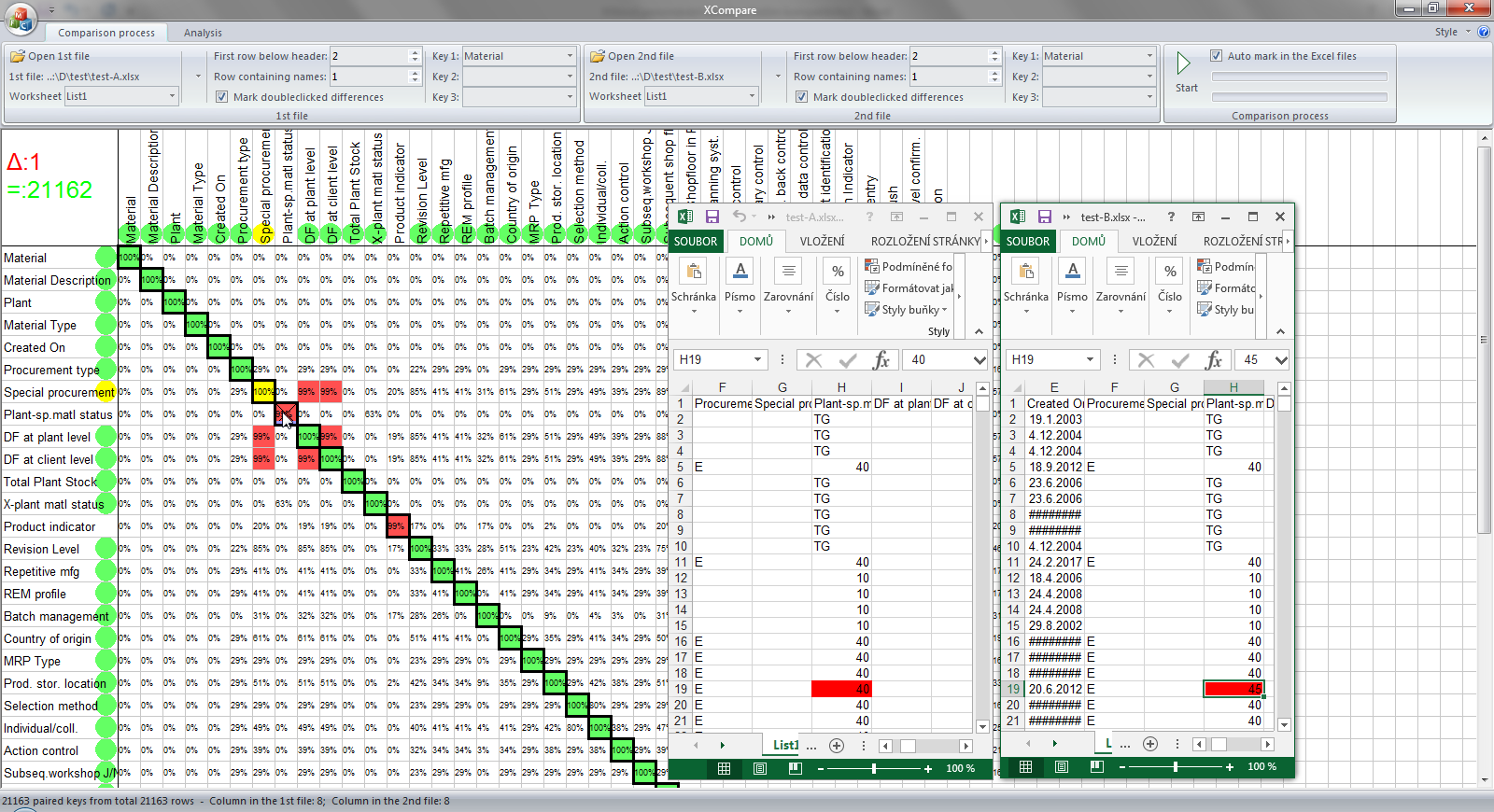
1. Porovnání tabulek obsahujících různý počet řádků, sloupců, různé názvy sloupců v hlavičce a rozdílné počty řádků hlavičky: V tomto extrémním případě (pro který ale byla popisovaná metoda vlastně vytvořena) je výsledkem práce programu matice shod porovnávaných sloupců. Stejně jako v ostatních případech jsou v hlavní pracovní oblasti nástroje zobrazeny hodnoty buněk této matice vyjádřené v procentech – tedy: 0% = žádná nalezená shoda, 100% = všechny nalezené hodnoty se shodují. Hodnoty jsou dále vizualizovány dle tohoto schematu (schema je součástí dialogu „about“ programu XCompare):



(Míra podobnosti, která je vyhodnocena jako „podezřelá“ je samozřejmě uživatelsky nastavitelná)

Zcela vlevo v pracovní oblasti jsou uvedeny názvy sloupců z prvního souboru, Nahoře jsou pak zobrazeny názvy sloupců druhého porovnávaného souboru. Názvy sloupečků jsou u každého souboru převzaty z řádku zvoleného uživatelem. Tento řádek lze kdykoli později změnit bez nutnosti zopakovat porovnávací výpočet. Barevné značky vyjadřují 100% shodu. Zeleně jsou označeny sloupečky, pro které byl nalezen 100% identický sloupec ve druhém souboru. Žlutá barva označuje prázdné sloupečky. Tyto dva případy jsou aplikací rozlišovány z praktických důvodů. Bez návodného popisu sloupečků (ev. bez detailní znalosti obsahu tabulek) je totiž obtížné zodpovědně rozhodnout, zda mají být oba sloupečky skutečně prázdné. Účelem žluté značky je tedy upozornění uživatele na potřebu podrobnější analýzy souvislostí dané dvojice prázdných sloupečků. Po vytvoření matice je možné pouhým poklepáním na vybranou buňku nechat označit barvou všechny nalezené odlišnosti protínajících se sloupečků přímo - buďto v jednom, druhém nebo i obou souborech. program lze také přepnout do analytického režimu. Ten nabízí nástroje pro práci s hlavní pracovní oblasti. Lze např. vypnout zvýrazňování buněk v pracovní oblasti dle různých kritérií. V diskutovaném případě je ale zřejmě daleko užitečnější funkce umožňující velmi rychle lokalizovat prověřovanou odlišnost přímo v Excelu. Za zmínku stojí také možnost nastavení barvy, která bude následně použita pro označení vybraných nalezených odlišností rovněž přímo v Excelu.



1. Porovnávání rozsáhlých tabulek obsahujících data v identicky pojmenovaných sloupcích (v hlavičce): V tomto případě je možné nechat nástroj označit nalezené odlišnosti přímo v porovnávaných souborech zcela automaticky. Využití této funkce nevyžaduje, aby pořadí sloupců ani řádků v souborech bylo stejné. Program ale označí pouze ty hodnoty (v jedné nebo obou tabulkách - podle nastavení), které se nacházejí ve sloupcích se stejnými názvy v obou tabulkách. Pokud tedy nejsou jména sloupců stejná, je tato automatická funkce z principu nepoužitelná.
2. **Metoda (zjednodušený popis)**

Předpoklad: V porovnávaných tabulkách je alespoň jeden sloupec, obsahující údaje, které jsou pro kombinaci těchto sloupců jedinečné. Alternativně lze spojit údaje z více sloupců tak, aby jejich kombinace načtená z jednoho řádku byla pro daný soubor jedinečná (např. kombinace SNr dílu a čísla dodavatele).

Před spuštěním hlavního algoritmu musí tedy být ověřena jedinečnost klíčových dat.

Pozn. k implementaci: program **XCompare** provádí toto ověřování velmi rychle před každým porovnáváním.

Postup práce hlavního algoritmu: Z prvního souboru je v každém řádku nejprve načten klíčový údaj. Tento klíčový údaj je následně vyhledán i ve druhém souboru. Pokud je nalezen, jsou porovnány údaje ve všech sloupcích prvního souboru s údaji ve všech sloupcích druhého souboru. Pokud je nalezena shoda, je inkrementována hodnota (zpočátku nulová) vyjadřující počet všech shod pro danou dvojici sloupců obou souborů. Tato hodnota je ukládána do odpovídajícího prvku (nacházejícího se na souřadnicích daných pořadím sloupců v jednom i druhém souboru) matice shod.

Algoritmus stejným způsobem postupuje dál, dokud neprojde všechny řádky prvního souboru a nespočítá nalezené shody ve všech sloupcích druhého souboru. Výsledkem zpracování hlavního algoritmu je pak matice všech nalezených shod.

V případě, že je aktivován požadavek na automatické označování neshod, je navíc při jakékoli nalezené neshodě prověřována shoda názvů prohledávaných sloupečků. Pokud je shoda názvů prokázána, jsou neshody hodnot označeny přímo ve zdrojových souborech.

1. **Další součásti návrhu:**

Zatím nedodanou součástí návrhu jsou komentované zdrojové kódy programu, popis postupu kompilace (lze např. sestavit 64bit verzi pro použití s velmi velkými soubory) a podrobný uživatelský návod k použití.

24.10.2017

Jiří Havlík