ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

Katedra systémového inženýrství

Unifikace klientů

Bakalářská práce

Vypracoval: Milan Vrbenský

Vedoucí bakalářské práce: doc. RNDr. Helena Brožová, CSc.

Zadání BP

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Unifikace klientů" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne datum odevzdání _____

Poděkování

Rád bych touto cestou poděkoval doc. RNDr. Heleně Brožové, CSc. za rady ohledně použití nejvhodnějších metod pro unifikaci.

Své rodině za trpělivost a podporu v době psaní této práce.

RNDr. Jakubovi Strnadovi, generálnímu řediteli České podnikatelské pojišťovny, a.s., Vienna Insurance Group za souhlas s testováním unifikace nad daty této pojišťovny.

Unifikace klientů

Clients unification

Souhrn

Tato práce se zabývá analýzou problematiky unfikace klientů v datvém skladu. Je zaměřena na analýzu možných přístupů a problémů s touto tématikou spojených. Součástí práce je i návrh jednoduché třístupňové unifikace s využitím rozhodovacího stromu a metody váženého součtu. Unifikace byla realizována v prostředí SAS a otestována na vzorku reálných dat. Kód pro unifikaci je součástí práce.

Summary

This paper deals with analysis of client's unification in data warehouse. It focuses on analysis of possible approaches and issues related to this topic. The paper also comprises a draft of simple three-stage unification using decision tree and the weighted summary. The unification was implemented in SAS environment and tested on a sample of real data. The code for unification is also part of the paper.

Klíčová slova: unifikace, unifikace klientů, rozhodovací strom, metoda váženého součtu, Saatyho metoda, SAS

Keywords: unification, clients unification, decission tree, weighted summary, Saaty's method, SAS

Obsah

1 Úvod

V dnešní době masového nasazení výpočetní techniky ve firmách není výjimkou, aby jedna firma používala více provozních systémů. Není neobvyklé ani elektronické zpracování dokumentů, automatické scanování a rozpoznávání psaného textu, různé asynchronní a synchronní služby pro komunikaci mezi systémy apod. V takovém prostředí je náročné, ne-li nemožné, udržet konzistenci informací o klientech firmy. Klient je přitom nedůležitějším vlastnictvím firmy. Po implementaci datového skladu ve firmě je tedy unifikace klientů první logickou nadstavbou.

Nový datový sklad České podnikatelské pojišťovny, a.s., Vienna Insurance Group (ČPP), jehož je autor vedoucím, se nachází přesně v této fázi. Tedy ve fázi, kdy je třeba z datového skladu začít čerpat relevantní informace jako podklad pro rozhodování v rámci společnosti. Jako podklad pro rozhodování ohledně kampaní pro oddělení marketingu, jako podklad pro rozhodování obchodního úseku ohledně vedení obchodní služby. Datový sklad by měl začít plnit svou úlohu centrálního úložiště a centrálního dodavatele správných dat. V této fázi se množí požadavky na základní reporty ohledně propojištěnosti jednotlivých oblastí, portfolia smluv klientů, chování jednotlivých klietů a jejich platební morálku, škodní průběhy apod. Všechny tyto reporty je však třeba vytvářet nad unifikovanými klienty. Snaha o tuto unifikaci má tedy vysokou prioritu.

Unifikaci klientů je možno řešit buď interně nebo nákupem některého komerčního produktu. Implementace komerčního produktu je však relativně dlouhodobá záležitost. Zároveň se jedná o poměrně velkou investici v řádu stovek tisíc až miliónů korun. Proto vznikla potřeba problém řešit interně.

Tato práce by měla pomoci realizaci interního řešení unifikace klietnů v datovém skladu ČPP jak na úrovni analýzy, tak v rámci konkrétní implementace unifikace na platformě SAS datového skladu.

2 CÍL A METODIKA

2.1 Cíl

Velké firmy dodávají sofistikované nástroje pro unifikaci klientů. Vlastní realizace této unifikace však bývá střeženým tajemstvím každé firmy.

Cílem práce je:

- vyplnit mezeru v dostupných informacích ohledně realizace konkrétní úlohy unifikace klientů a navrhnout jednoduchou a funkční metodu pro tuto úlohu
- programově realizovat unifikaci v datovém skladu na technologii SAS
- otestovat a vyhodnotit navrženou unifikaci na vzorku dat České podnikatelské pojišťovny, a. s., Vienna Insurance Group

2.2 Metodika

Vzhledem k nedostatku informací ohledně skutečné realizace unifikace klientů vznikla tato práce zejména na základě konzultací s lidmi z firmy SAS Institute ČR, s.r.o. a BQS Consulting, s.r.o. a na základě znalostí získaných na školeních pořádaných firmou Adastra s.r.o. Konstrukce unifikace vychází z koncepce a ze struktury nově vznikajícího společného datového skladu pro Českou podnikatelskou pojišťovnu, a.s., Vienna Insurance Group a Kooperativu, pojišťovnu, a.s., Vienna Insurance Group.

Zdrojový kód pro testování navržené metody unifikace vznikal průběžně v době zpracovávání tématu do bakalářské práce. Zdrojový kód není optimalizován a nesplňuje všechny požadavky pro reálné nasazení. Je určen pouze k otestování navržené metody. Testování bylo vzhledem k náročnosti provedeno na vzorku dat.

Vlastní unifikace dat je vlastně pokus o svázání dvou a více klientských záznamů náležících jednomu klientovi. Rozhodování je prováděno automaticky

na základě předem definovaných pravidel. Řešení tedy směřuje k implementaci modelu vícekriteriálního rozhodování. Tento model je implementován v posledních dvou krocích unifikace. Tyto kroky jsou netechnické a jedná se o vyšší míru unifikace. V třetím kroku unfikace byla použita metoda rozhodování pomocí *rozhodovacího stromu*. Ve čtvrtém kroku unifikace pak *metoda váženého součtu*. Váhy byly stanoveny *Saatyho metodou*.

3 POPIS METOD VHODNÝCH PRO PROCES UNIFIKACE KLIENTŮ

Definice unifikace klientů

Unifikace klientů je proces identifikace každého jednotlivého klienta firmy a dohledání všech dpulicitně se vyskytujících záznamů o tomto klientovi v rámci všech dat ve firmě dostupných.

Důvody vedoucí k potřebě unifikace v pojišťovnictví

Trh ve finanční sféře, zvláště v pojišťovnictví, by se dnes dal označit slovem „nasycený“. Posledním velkým přerozdělením klientů byla změna zákona v roce 2000, kdy došlo ke zrušení monopolního postavení České Pojišťovny, a. s. v oblasti pojištění povinného ručení vozidel. Nabídky jednotlivých pojišťoven jsou dnes velice podobné a získat nové zákazníky v takovém prostředí je nákladné. Prostředí tak vyvíjí značný tlak na subjekty zde vystupující. Metod, jak získat nové zákazníky nebo udržet stávající a přitom minimalizovat náklady, je několik. Na některé se podíváme detailně.

Ekonomicky nejnevýhodnější možností oslovení zákazníků, ať již stávajících nebo potencionálních, je *plošná marketingová kampaň*. Náklady na takovou kampaň jsou vysoké, účinnost je diskutabilní. Většinou však ne příliš velká.

Cílená kampaň na „cross selling“ je speciální druh kampaně, ve které se pojišťovna snaží oslovit stávající klienty a rozšířit počet smluv s klientem uzavřených. Klient je tak krytý na širší okruh rizik, klesají jednotkové náklady na zpracování dat tohoto klienta, zvyšuje se pravděpodobnost, že klient další smlouvu uzavře opět u této pojišťovny a v neposlední řadě vzrůstá náročnost přechodu klienta k jiné pojišťovně.

Cílená kampaň na získání nových zákazníků spočívá ve vytvoření zajímavého pojistného produktu, stanovení takové ceny nebo nastavení takových podmínek, které nové zákazníky osloví. Informace o potřebách klientů lze získat na základě analýzy

stávajících klientů.

Chce-li však pojišťovna maximalizovat zisk, nemůže se soustředit pouze na získání a udržení klientů. Důležitá je i správa a analýza stávajících klientů. Existují situace, ve kterých nemusí být udržení stávajícího klienta nebo získání nového klienta žádoucí. Jedná se zejména o klienty spojené s pojistnými podvody nebo například pojištění velice rizikového klienta. V tomto případě je používána tzv. *segmentace klientů*. Segmentací klientů rozumíme rozdělení klientů na jednotlivé skupiny. Přístup k těmto skupinám je pak různě odstupňován a jsou například definovány různé sazby. Příkladem může být pojištění léčebných výloh v zahraničí, kde se liší výše pojistného u klientů jedoucích na dovolenou do lázní a klientů jedoucích na dovolenou za účelem potápění nebo vysokohorské turistiky. Zde dochází k jednoduché segmentaci klientů na základě výše rizika.

Všechny tyto metody mají společnou potřebu. A tou je *znalost klientského portfolia* a *čistota těchto dat*.

Obecné metody unifikace klientů

CRM

Mnoho firem dnes přechází na takzvané CRM systémy, tj. „Řízení vztahů se zákazníky“ (zkratka pochází z anglického „Customer relationship management“). Tyto systémy jsou navrženy s důrazem na klienta. Informace o klientech jsou uloženy v tomto systému a ostatní systémy tyto informace využívají. CRM systémy se tak mohou řadit mezi možné metody unifikace klientů. Mají však jednu velkou nevýhodu – snadno je lze použít jen na *nové* klienty. V případě konverze informací ze stávajících systémů do CRM se však do nového systému dostanou chybné, neúplné, případně duplicitní informace a tento systém od počátku neplní funkci tak, jak by měl.

Vlastní provoz takového systému s sebou však nese rizika ztráty čistoty dat.

Jedná se jak o rizika technického rázu - tedy o špatně nastavené procesy zpracování klientských dat, kdy může dojít k duplikování dat, o chybně nastavené nebo neúplně definované kontroly dat při zadávání, nesprávně nastavené automatické importování dat do systému apod. A také o rizika spojená s běžným životem klienta, například o změnu adresy klienta, kterou klient nenahlásí apod.

Dalším problémem těchto systémů je cena implementace. Zavedení takového systému například pro pojišťovnu je poměrně náročný, drahý a dlouhodobý úkol.

Zvyšování kvality klientského portfolia

Mezi metody unifikace lze zařadit i *proces zvyšování kvality klientského portfolia*. Ať již firma disponuje, zavádí či nezavádí CRM, je pro ni důležité znát co nejvíce správných a aktuálních informací o klientech. A informace jsou, jak již bylo řečeno, zpravidla nějakým způsobem zkreslené. Zvyšování kvality klientského portfolia je vlastně proces sběru dostupných informací o klientovi a jejich zaznamenání. Ani tato cesta však není jednoduchá. Sběr informací musí být nějak řízen. Musí tedy existovat systém, do kterého je nutné změny zapisovat. Data se v tomto případě opravují „zpětně“ například po telefonickém rozhovoru s klientem. Systém data o klientech průběžně aktualizuje, nicméně to neznamená, že pojišťovna má aktuální data o klientech. Tento systém pouze pomáhá dlouhodobě udržet kvalitu klientských dat. Navíc vznikají další náklady se školením pracovníků na užívání tohoto systému apod. Ani tato cesta tedy není jednoduchá a ani v tomto případě nelze vyloučit chyby v datech. Přesto pojišťovna ke svému rozhodování potřebuje co nejpřesnější data.

Unifikace klientských záznamů z provozních systémů

Aby bylo možno data o klientech plně využít pro účely marketingu, segmentace, odhalování pojistných podvodů apod. je třeba data „vyčistit“. Tj. nastavit takový proces, při kterém dojde k identifikaci konkrétních klientů ve všech systémech a jejich unifikaci v rámci celé firmy. Tomuto procesu se říká *unifikace klientů*.

Kdyby existovaly ideální systémy a ideální uživatelé, byla by unifikace jednoduchá. Všechny informace by byly správné a kompletní. Realita je však jiná. Každý systém zpravidla uchovává o klientech jiná data. Data do systému vkládají lidé, případně jsou data zpracovávána elektronicky pomocí rozpoznávání písma. A to znamená i chyby. Některé systémy implementují více či méně pokročilou správu chyb a kontrolují vstupující údaje. Nicméně ani tak se nevyhneme nedostatkům v datech.

Tyto nedostatky lze nejsnadněji řešit v datovém skladu. Datovým skladem se rozumí centrální uložiště informací z celé firmy. Samozřejmě i informací o klientech. Datový sklad zpravidla obsahuje vlastní strukturu dat o klientech. Do této struktry se pak v pravidelných intervalech transformují data ze všech dostupných systémů. V datovém skladu tak najdeme klienty ze všech systémů, a to ve *shodné struktuře*. Datový sklad se tak stává ideálním místem pro proces unifikace.

Je důležité si však uvědomit, že proces unifikace není vždy jednoznačný. Pokud neustoupíme z nároků na přesnost dat, nepovede se nám provést úspěšně unifikaci. Naopak nastavení příliš volných kritérií pro unifikaci povede k přílišné unifikaci a ke ztrátě informací. Jinak řečeno každý krok unifikace vede více či méně ke zvýšení možnosti chybné unifikace, a tak ztráty dat. Toto nastavení je věcí každé firmy a je třeba míru unifikace zvážit. Ke správnému návrhu, do jaké míry můžeme ustupovat z detailu dat, musíme znát potřeby a specifika firmy a cíl konkrétní úlohy a odhadnout ji. Nastavení míry unifikace je tak individuální nejen pro každou firmu, ale do jisté míry i pro každou úlohu. Je jasné, že při použití vyšší míry unifikace u marketingové akce nemusíme oslovit několik klientů (budeme dva klienty považovat za jednoho) a následky nebudou nijak vážné. Naopak se minimalizuje pravděpodobnost vzniku negativní reklamy v případě oslovení jednoho klienta dvakrát. Použití této míry unifikace například v procesu vymáhání pohledávek však bude mít katastrofální následky vedoucí k medializaci a k negativní reklamě celé firmy. Proto je unifikace realizována tak, aby bylo možno pro konkrétní úlohu použít odpovídající míru unifikace.

3.1 Metody unifikace klientů v datovém skladu

Realizace unifikace v datovém skladu může být uskutečněna nejrůznějšími postupy a data mohou být uložena nejrůznějším způsobem. Existují však přibližně dva základní způsoby provádění unifikace a dva základní způsoby uložení dat.

Hodnocení všech dvojic klientů

Výsledkem této metody je výpočet váhy určující míru pravděpodobnosti, že dva záznamy patří ve skutečnosti jednomu klientovi. Jedná se tedy o sekvenční testování každé dvojice záznamů o klientech a určení výsledné pravděpodobnosti. Při výpočtu této váhy samozřejmě musíme brát v úvahu všechna rizika v čistotě dat atd.

Výhody:

- Lze snadno uchovávat historicky dosažená hodnocení a vyhodnocovat změny v systémech vedoucí ke zlepšení (zhoršení) informací o klientech.
- Lze jednoduše implementovat inkrementální načítání dat.

Nevýhody:

- Rychlost zpracování (vyhodnocují se všechny dvojice záznamů).
- Možnost vzniku **křížových závislostí.** [1]

Použití master a slave záznamů

V rámci této metody dojde k rozdělení záznamů na tzv. Master a Slave. Master

1 Jedná se o případ, kdy pro konkrétní úroveň unifikace dojde ke zkřížení unifikací. Může nastat případ, kdy záznam klient2 odpovídá podle unifikace záznamu klient1. A zároveň klient3 odpovídá klientu2. Nicméně vazba mezi záznamy klient1 a klient2 nedosahuje požadovaného skóre.

záznamy jsou tvořeny množinou klientských záznamů, které lze v rámci možností považovat za správné a úplné. Každý z těchto záznamů tedy například obsahuje správně vyplněné rodné číslo vyhovující testu modulo 11, vyplněné jméno a příjmení odpovídající některé položce v seznamu jmen a příjmení v ČR a správně vyplněnou adresu i s přiřazením kódu odpovídající číselníku adres. Dodatečným požadavkem na master záznamy je pak unikátnost klientů v celé množině. Většinou tedy požadavek na unikátnost rodného čísla v celé množině záznamů. Slave záznamy obsahují všechny ostatní záznamy. V rámci této metody pak dochází k porovnávání pouze mezi master a slave záznamy.

Výhody

- Rychlost zpracování (porovnávají se vždy pouze záznamy master a slave).
- Nemůže dojít ke křížové závislosti.

Nevýhody

- Omezení variant unifikace na předem daný počet stupňů.
- Obtížnější implementace inkrementálního načítání.
- Obtížnější implementace historického pohledu na data.

3.2 Metody uložení unifikace v datovém skladu

Vazební tabulka obsahující skóre

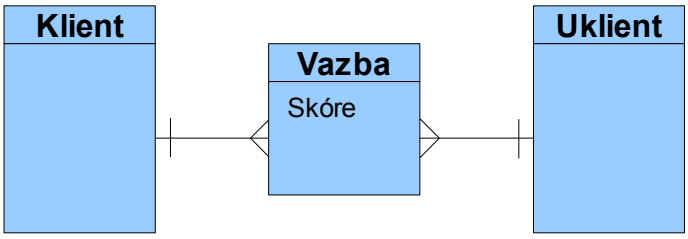
Kromě tabulky klientů (Klient) a tabulky unifikovaných klientů (Uklient) vzniká vazební tabulka (Vazba) spojující záznamy těchto tabulek mezi sebou. Vazební tabulka obsahuje vazbu mezi záznamem z tabulky Klient a Uklient a pravděpodobnost s jakou tyto záznamy náleží jednomu klientovi. Hodnota pravděpodobnosti je většinou vyjádřena v bodovém ohodnocení. Konceptuální model vyjadřuje Obrázek 1.

Výhody

- Pro každou praktickou úlohu, v níž je třeba použít unifikovaná data, lze použít specifickou míru unifikace.
- Lze jednoduše vyhodnocovat historické změny.

Nevýhody

- Nutnost dalších algoritmů před nasazením unifikace pro konkrétní úlohu.
- Vyšší možnost špatného odhadu potřebné míry unifikace pro konkrétní úlohu.
- Možnost chyb při použití různých měr unifikace pro podobné úkoly a tím vznik neporovnatelných výsledků. Toto se týká zvláště statistických pohledů na data.



Obrázek 1: Uložení skóre

Vazební tabulka s předpřipravenými hodnotami unifikace

Kromě tabulky klientů (Klient) a tabulky unifikovaných klientů (Uklient) vzniká opět vazební tabulka (Vazba). V této tabulce jsou předpřipraveny vazby pro různé stupně unifikace. Tabulka tak obsahuje například tři sloupce obsahující cizí klíč do tabulky unifikovaných klientů pro tři různé stupně unifikace (Unifikace1, Unifikace2, Unifikace3). Nejnižší míru unifikace tak lze přímo použít např. na správu pohledávek, střední pro analýzy, nejvyšší unifikaci pro marketingové akce.

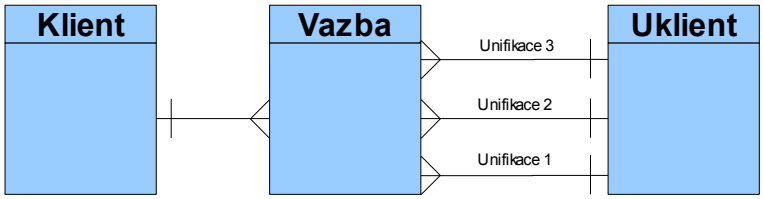
Konceptuální model vyjadřuje Obrázek 2.

Výhody

- Jednoduchá implementace dat v modelu.
- Zjednodušené rozhodování o použití míry unifikace pro danou úlohu.
- Jednoduché použití, srozumitelnost.

Nevýhody

- Obtížné sledování historie.
- Menší flexibilita modelu.



Obrázek 2: Předpřipravené hodnoty unifikace

4 NAVRŽENÁ METODA UNIFIKACE KLIENTŮ

V rámci této práce byla použita metoda master a slave záznamů pro unifikaci. Data jsou ukládána do vazební tabulky s předpřipravenými mírami unifikace. Následující informace se vztahují k této kombinaci unifikace a uložení dat.

4.1 Seznam atributů klientů

- Příjmení
- Jméno
- Rodné číslo
- Kód UIR-ADR[2]
- Město
- Ulice
- PSČ

4.2 Popis jednotlivých dostupných atributů a vyhodnocení rizik

Příjmení

Příjmení by mělo být nedílnou součástí každého občanského klienta. U korektně navrženého systému by se klienti bez zadaného příjmení neměli vyskytovat.

Kontrola příjmení je ale poněkud problematická. Některá omezení lze definovat přímo. Například požadavky na absenci numerických znaků v příjmení, absenci mezer apod. Dalším možným požadavkem je pak kontrola vůči databázi

2 Územně identifikační registr adres

příjmení z posledního sčítání lidu. Tato data zveřejňuje na svých internetových stránkách Český statistický úřad. Nicméně čistota dat v těchto údajích není vysoká. Cizinci pak v těchto datech být nemusí, což by mohlo vést k chybným výsledkům.

Z databáze příjmení lze také vyčíst četnost výskytu jednotlivých jmen a celkový počet unikátních jmen v ČR (60 227 pro muže, 63 602 pro ženy). Z těchto údajů lze odhadnout váhu tohoto atributu v rámci unifikace. Data lze využít i v rámci čištění dat například na odhad pohlaví klienta apod.

Příjmení je však zdrojem i některých záludných chyb. V první řadě se jedná o textový údaj. Při vyplňování může dojít k překlepu nebo ke špatné interpretaci čteného znaku v průběhu rozpoznávání při automatickém scanování dat. V lepším případě dojde k chybě v příjmení. V horším případě pak dojde k záměně jednoho příjmení za jiné. Druhou oblastí chyb je poměrně častá změna příjmení hlavně u žen v důsledku sňatku.

Jméno

Jméno je velice podobné atributu příjmení. Také by mělo být nedílnou součástí občanského klienta. Kontroly na jméno lze definovat obdobným způsobem. Rozdíl je zde hlavně v počtu jmen používaných v rámci ČR. Počet jmen mužů je 4 227, žen 3 850. Je tedy vidět, že váha tohoto atributu by měla být menší nežli u příjmení.

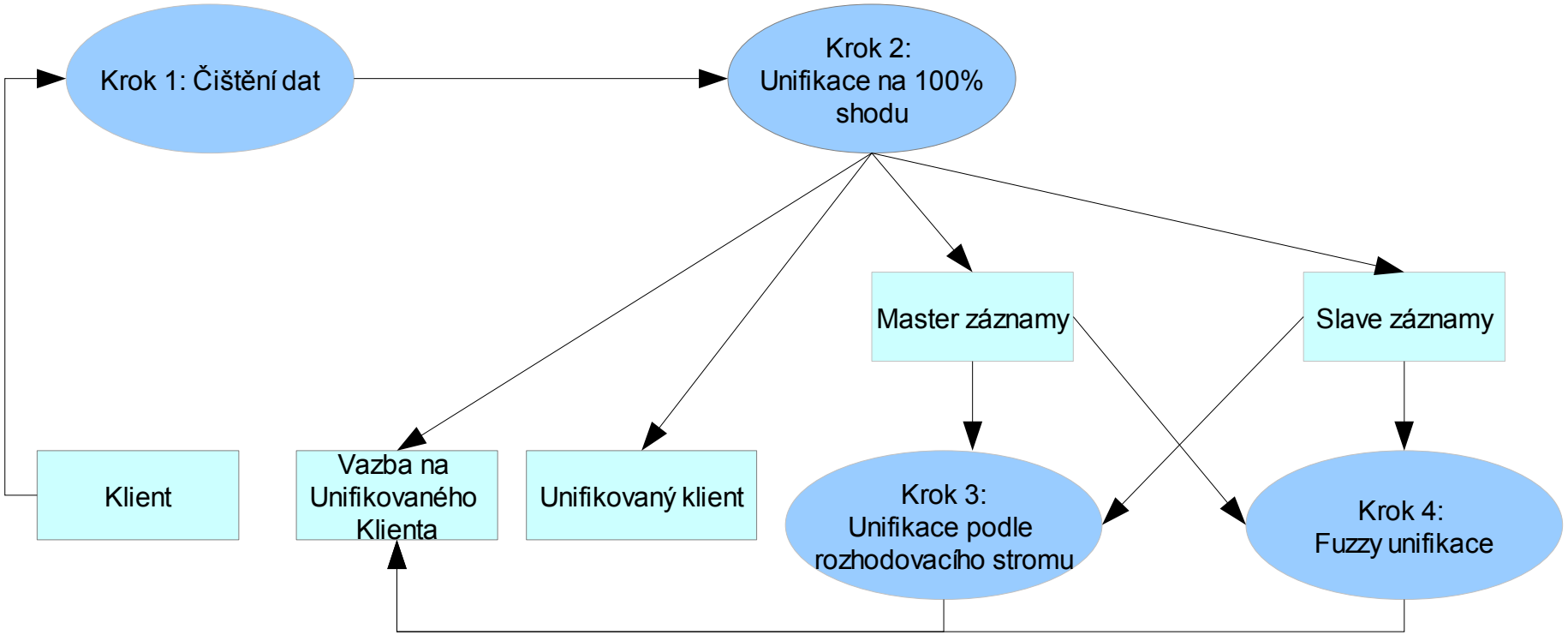
Zdroje chyb jsou shodné s příjmením. Nicméně je zde velká výhoda v minimální pravděpodobnosti změny jména klienta.

Rodné číslo

Rodné číslo je další nedílná součást občanského klienta, nicméně existují výjimky. Jedná se zejména o cizince, kterým nebylo rodné číslo přiděleno. V takovém případě je nahrazováno datem narození případně zcela chybí.

Chyba v rodném čísle občanů ČR narozených po roce 1953 je velice snadno

odhalitelná. Jedná se o propočet modulo 11. Pro proces unifikace je však možné využít i případně neúplně zadané rodné číslo. Z prvních 6ti číslic lze vypočítat věk klienta a jeho pohlaví. Tyto ukazatele pak lze použít pro některý krok unifikace. I zde ale existují skrytá rizika chyb. V případě ČPP se jedná o přidělování fiktivních rodných čísel u některých druhů pojištění. Tato fiktivní rodná čísla je třeba identifikovat a tuto hodnotu nepoužívat pro následnou unifikaci.



Obrázek 3: Schéma průběhu unifikaceK

Adresa

Adresa je soubor několika atributů určující trvalou adresu klienta. V případě ČPP je adresa složena z města, ulice (ulice + číslo popisné), PSČ a kódu UIR-ADR. Ministerstvo práce a sociálních věcí udržuje ve spolupráci s obecními úřady „Územě identifikační registr adres“ (ÚIR-ADR). V rámci provozního systému pojišťovny již v případě správného zapsání adresy do systému dojde k navázání této adresy na registr a k zápisu kódu do adresy klienta.

Chyba je velice snadno odhalitelná. Identifikaci chyby lze navázat na existenci vazby do UIR-ADR. Tímto způsobem nelze odhalit záměnu adres, ale lze ověřit správnost adresy jako takové. Toto je samozřejmě omezeno aktuálností zveřejňovaných informací. Může existovat adresa, která zatím v registru není zavedena. Pravděpodobnost je ale minimální a chyba se automaticky opraví při aktualizaci dat. Důležitá je však skutečnost, že chybně napsaná část adresy způsobí chybu v přiřazení dat do číselníku a kód uir-adr nelze přiřadit. Zahodit kompletní adresu například při překlepu v čísle popisném je ale zbytečné. Proto je kód UIR-ADR brán jako identifikátor zda zadaná adresa existuje, ale v rámci unifikace se pracuje i s položkami město, ulice a PSČ. U klientů lze vypozorovat některé vzorce chování. Lidé se většinou stěhují v rámci města, případně okresu apod. Tyto informace lze také využít.

4.3 Postup navržené unifikace

Postup unifikace ukazuje Obrázek 3. V první řadě je provedeno čištění dat, které je popsáno v předchozí kapitole. Následuje 100% unifikace. Tento krok unifikace je nejnižší míra unifikace a zároveň definuje záznamy v tabulce unifikovaných klientů. Z technických důvodů se v tomto kroku rozdělují záznamy klientů na master a slave. Následuje unifikace pomocí rozhodovacího stromu (vyšší úroveň unifikace) a fuzzy unifikace (nejvyšší míra unifikace).

Data o klientech jsou získávána z datového skladu, kde jsou dostupné údaje ze všech provozních systémů, a to ve shodné struktuře.

4.4 Krok 1 – čištění dat

Před vlastní unifikací je třeba data očistit, případně doplnit z dostupných datových zdrojů další informace. Čištění dat však není součástí této práce. Tento krok byl proto omezen na minimum.

Čištění dat bylo omezeno na následující atributy:

- Identifikaci občanských klientů
- Rodné číslo

Identifikace občanských klientů

Data v datovém skladu jsou kombinací údajů o podnikatelských klientech a občanských klientech. Je tedy třeba je rozdělit. Rozdělení provedeme podle atributu určujícího typ klienta a dále se budeme zabývat pouze unifikací občanských klientů. Unifikace podnikatelských klientů by probíhala podobným způsobem. Rozdíl by byl samozřejmě v pravidlech pro identifikaci jednotlivých klientů a v nastavení kritérií za jakých je možno dva klientské záznamy unifikovat.

Rodné číslo

Kontrola rodného čísla je prováděna pomocí kontroly „modulo 11“. Jedná se o test, zda zbytek po dělení rodného čísla číslem 11 je nulový, případně je roven 10. V rámci této unifikace je ale kontrola prováděna pomocí makra vytvořeného a dodávaného společností SAS ČR. Makro kontrolu provede včetně výjimek u rodných čísel s datem narození před rokem 1953, kdy nebyla kontrola modulo 11 zavedena. V rámci kontroly je také využívána další funkce tohoto makra umožňující získat datum narození a věk osoby i z ne zcela kompletního rodného čísla. Tento údaj samozřejmě není úplně

důvěryhodný, nicméně jeho informační hodnota je využívána v dalších úrovních unifikace.

Pravděpodobnost výskytu chybně vloženého rodného čísla, které by odpovídalo kontrole na modulo 11 je téměř nulová. Proto pro unifikaci rozdělíme data na data s rodným číslem ověřeným podle modulo 11 a na ostatní. Mezi ostatní data dále zařadíme klienty s fiktivním rodným číslem. Jedná se o číselnou řadu desetimístných rodných čísel začínajících 00.

4.5 Krok 2 – unifikace na 100% shodu, rozdělení dat na master a slave záznamy

Jedná se vlastně o nejvyšší stupeň unifikace. Ke sloučení dvou a více záznamů, u kterých existuje shoda ve všech faktových atributech unifikovaných záznamů. V unifikaci občanských klientů se jedná o jméno, příjmení, rodné číslo, uir-adr, ulici, město, PSČ. Do této unifikace však nevstupují technické informace záznamů. V našem případě do unifikace nevstupuje atribut ID, který je tvořen automaticky generovaným pořadovým číslem a slouží pro identifikaci záznamu v databázi. Tento druh unifikace má význam zvláště v prostředí s několika provozními systémy, a to k unifikaci klientů napříč těmito systémy. Lze totiž předpokládat, že jeden a ten samý klient se z nějakého důvodu vyskytuje ve dvou a více systémech zároveň. Předpokládáme-li snahu každé organizace o co nejvyšší čistotu dat, existuje relativně velká pravděpodobnost stoprocentní shody záznamů o daném klientovi v těchto systémech.

Důležitou vlastností této unifikace je skutečnost, že *nedochází ke ztrátě informace*. Zároveň**,** protože se jedná o krok unifikace s největším objemem výstupních dat, nám tento krok vytvoří cílová data pro tabulku „unifikovaných klientů“ (Uklient).

Skutečná realizace tohoto kroku se jeví jako jednoduchá. Lze ji realizovat jednoduchým SQL příkazem. Skutečnost je však složitější. Problémem je potřeba zachování technického identifikátoru a vytvoření vazebních dat mezi původními

záznamy a unifikovanou sadou záznamů. Systém SAS naštěstí má nástroje jak tohoto docílit. Krok je realizován pomocí procedury data. Data jsou seřazena sestupně podle všech atributů, a to včetně primárního klíče. Tedy seřazené sestupně podle rodného čísla, jména, příjmení, ulice, města, psč, uir-adr. Ze shodných záznamů je do výstupu zahrnut jen poslední záznam. Tím dostaneme do cílové tabulky „nejnovější“ hodnoty o klientovi. Zároveň je vytvářena vazební tabulka.

Unifikace je náročný proces na čas procesoru. Proto je třeba minimalizovat operace potřebné pro unifikaci klientů. Řešením je tvorba množiny tzv. master a slave záznamů. Jedná se o co možná nejpřesnější údaje o klietnovi. Tyto záznamy jsou samozřejmě vybírány z dat z unifikace na 100%.

Nároky na tato data se ale liší. Předně musíme zařídit jedinečnost rodného čísla v rámci sady master záznamů. Mezi master záznamy jsou řazeny pouze záznamy se správně vyplněným RČ, vyplněným jménem a příjmením a vazbou na UIR-ADR.

Záznamy, které z nějakého důvodu nelze zařadit mezi master záznamy se stávají slave záznamy. V následujících krocích pak budeme porovnávat master a slave záznamy mezi sebou a hledat vazbu ze slave záznamu na master.

4.6 Krok 3 – unifikace podle rozhodovacího stromu

Do této unifikace již vstupují i záznamy, které nejsou v množině master záznamů. Jedná se tedy již o 1. úroveň unifikace s přihlédnutím k možnostem některých chyb. Nicméně v tomto kroku nedochází k unifikaci zahrnující pokus o opravu dat. Tento krok unifikace je realizován pouze logikou přímo nad daty z datového skladu.

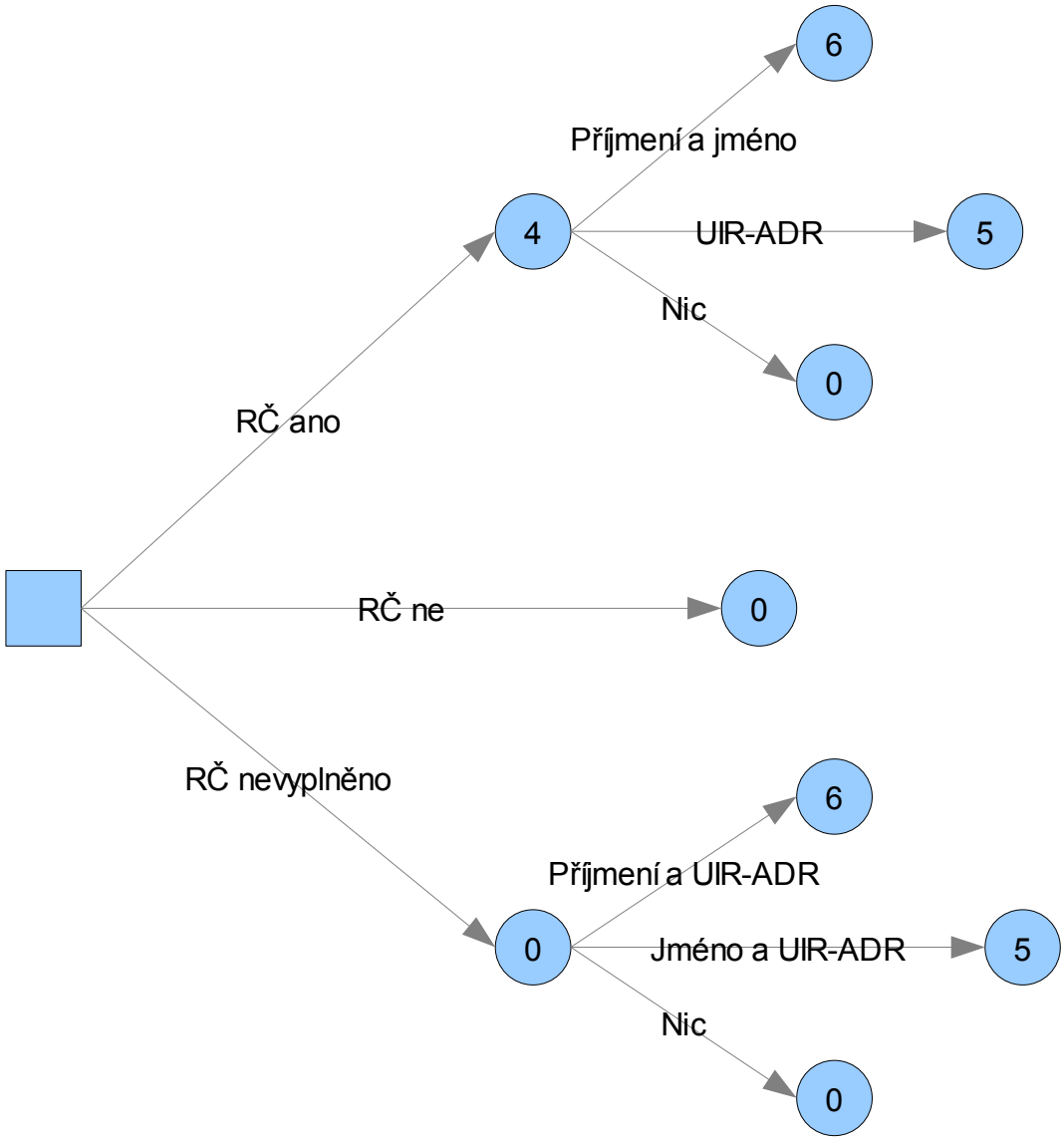
Kritéria pro to, aby bylo možno údaje dvou klientů považovat za údaje jedné osoby jsou uvedeny v Tabulka 1. V tabulce jsou zahrnuta i bodová hodnocení jednotlivých variant shody. Ta jsou využívána k určení nejvhodnějšího kandidáta z množiny master záznamů pro unifikaci. Důvodem je reálná možnost identifikace několika kandidátů z množiny master záznamů pro unifikaci. V takovém případě je

vybírán kandidát s nejvyšším bodovým hodnocením. V případě shodného bodového hodnocení je vybrána vazba na nejvyšším pořadovém čísle z master záznamů.

RČ	x	x		
Příjmení	x		x	
Jméno	x			x
Uir-adr		x	x	x
Ulice				
Město				
PSČ				
Věk				
Pohlaví				
Hodnocení	10	9	5	5

Tabulka 1: Kritéria pro hodnocení klientů ve 2. kroku

Krok je z praktických důvodů jednoduché implementace realizován pomocí rozhodovacího stromu. Jeho schéma je Obrázek 4: Krok 3 unifikace - rozhodovací strom.



Obrázek 4: Krok 3 unifikace - rozhodovací strom

4.7

4.8 Krok 4 – fuzzy unifikace

Jedná se o unifikaci s přihlédnutím k možnému výskytu překlepů apod. Jde o unifikaci klientů, pomocí které lze unifikovat i neúplné či chybné klientské záznamy v systému. Tato metoda poskytuje největší volnost a variabilitu v přístupu k realizaci.

Jedná se například o výpočet hodnocení jména a jeho významu vzhledem k ostatním atributům klienta. Ze statistických dat lze dohledat četnost výskytu konkrétního jména v rámci České republiky. Z této informace by bylo možné identifikovat četnost výskytu konkrétního jména klienta a podle toho vypočítat váhu tohoto atributu vzhledem k ostatním atributům klienta. Je jasné, že pradvěpodobnost shody u jména „Jiří“ (319 593 nositelů) je řádově vyšší než pravděpodobnost shody u jména „Ludislav“ (5 nositelů). Totéž platí o příjmení. Fuzzy unifikaci lze tedy realizovat například s dynamickým propočtem vah jednotlivých atributů klienta pro každého klienta odděleně a zpřesnit tak vlastní unifikaci. Nicméně je třeba si uvědomit, že vlastní realizace takového propočtu je výkonově náročná a je třeba zvážit vzniklý užitek. Ve stávající verzi unifikace nebyl tento algoritmus použit.

Vlastní určení zda se jedná o záznamy jednoho klienta již nelze vyřešit pomocí rozhodovacího stromu. Do procesu rozhodování vstupuje i vzdálenost dvou řetězců od sebe. Navíc je třeba zohlednit narůstající pravděpodobnost chyby v závislosti na rostoucí délce řetězce.

```
data test;
   infile datalines missover;
   input string1 $char8. string2 $char8. modifiers $char8.;
   result=complev(string1, string2, modifiers);
   datalines;
12345678123456781234567 8
Vrb     Vrbc    iln
Vrb     Vr      iln
Vrb     vRbc    iln
 Vrb    Vrb     iln
Vrb     'Vrb'n  iln
Řř      řŘ      iln
Milan   Tom     iln
;

proc print data=test;
run;
```

Výpis 1: Testovací kód Levenshteinovy editační vzdálenosti

Metoda je realizována metodou váženého součtu. Hodnotí se shoda každého

atributu. V rámci každého atributu je pak propočítáváno procento shody mezi master a slave záznamem. Pro propočet vzdálenosti dvou řeťezců je využíván algoritmus pro výpočet Levenshteinovy editační vzdálenosti. Funkce je standardní součástí SAS. Funkce vrací číslo určující nejnižší počet změn nutných k transformaci jednoho řetězce na druhý. Výpis 1 obsahuje testovací kód této funkce. Výpis 2 pak výsledky testu. Z výsledků je patrné, že funkce pracuje podle předpokladu a umí pracovat i s českými znaky.

Obs	string1	string2	modifiers	result
1	12345678	12345678	12345678	0
2	Vrb	Vrbc	iln	1
3	Vrb	Vr	iln	1
4	Vrb	vRbc	iln	1
5	Vrb	Vrb	iln	0
6	Vrb	'Vrb'n	iln	0
7	Řř	řŘ	iln	0
8	Milan	Tom	iln	5

Výpis 2: Výstup testovacího kódu Lavenstainovy editační vzdálenosti

U jednotlivých atributů je třeba ohodnotit jejich význam. K tomu je třeba provést analýzu dostupných informací.

Rodné číslo

I nadále je rodné číslo nejdůležitějším rozlišovacím znakem mezi klienty. I v tomto kroku jsou tedy záznamy rozděleny v první řadě podle rodného čísla na ty, kde

je rodné číslo správně vyplněno a na ostatní. Porovnávat vzdálenost rodného čísla master a slave záznamu je však bezpředmětné. Je třeba si uvědomit, že rodná čísla jsou přidělována jako řada. Proto by hodnota Levenshteinovy editační vzdálenosti mezi dvěmi rodnými čísly byla rovna 2. Rodné číslo je tedy použito pouze pro rozdělení slave záznamů na dvě množiny. Na záznamy s vyplněným rodným číslem a na záznamy s nevyplněným rodným číslem. Pro tyto dvě množiny jsou pak definovány jiné aspirační úrovně.

Jméno

Je poměrně snadné zjistit (například zde: http://www.mvcr.cz/clanek/cetnost-jmen-a-prijmeni.aspx) počet jmen používaných v ČR. Nicméně data jsou poněkud zkreslena. Jednoduchou analýzou konkrétních údajů uváděných ve výpisu lze zjistit, že celkový počet jmen v ČR (včetně cizinců) je 47 005. Avšak pouze 11 765 jmen má více než jednoho nositele. Pouze 7 150 jmen nosí více než 2 osoby, 5 261 více než 3 osoby. Empirickým průzkumem dat zjistíme, že data s nízkým počtem nositelů jsou buď špatně zadaná (například obsahují čísla apod.) nebo jsou to složená jména (například KURT EMIL) anebo jsou to jména cizinců. Vzhledem k tomu, že většinu klientů ČPP, nad jejímiž daty byla unifikace ověřována, jsou Češi, dopustili bychom se použitím celkového počtu jmen poměrně zásadní chyby (data o příjmení nejsou dostupná v rozdělení na ČR a cizinci). Je třeba si uvědomit, že musíme ohodnotit význam tohoto atributu vzhledem k ostatním atributům klienta. Použitím vysokého čísla bychom neúnosně nadhodnotili význam jména a vnesli bychom do unifikace chybu. Zvolili jsme tedy kompromis a do návrhu započítáme pouze jména, jež nosí alespoň pět osob v České republice. Tedy hodnotu 4 227 pro muže a 3 850 pro ženy.

Příjmení

Informace o počtech příjmení v ČR získáme na shodném místě. Jak již bylo zmíněno nejsou tato data dostupná v dělení na ČR a cizinci. Proto jsme museli z důvodu

porovnatelnosti používat i u příjmení souhrnná data. Z hlediska porovnatelnosti proto také aplikujeme shodné pravidlo pro počet. Tady do unifikace vložíme pouze ta příjmení, kde počet nositelů je alespoň 5. Tedy pro muže 60 227, pro ženy 63 602.

Ulice

Počet ulic v ČR lze zjistit z údajů územně identifikačního registru adres. Data jsou dostupná v datovém skladu ČPP. Z něho byla získána informace, že v ČR se nachází 27 413 ulic s různým názvem.

Město

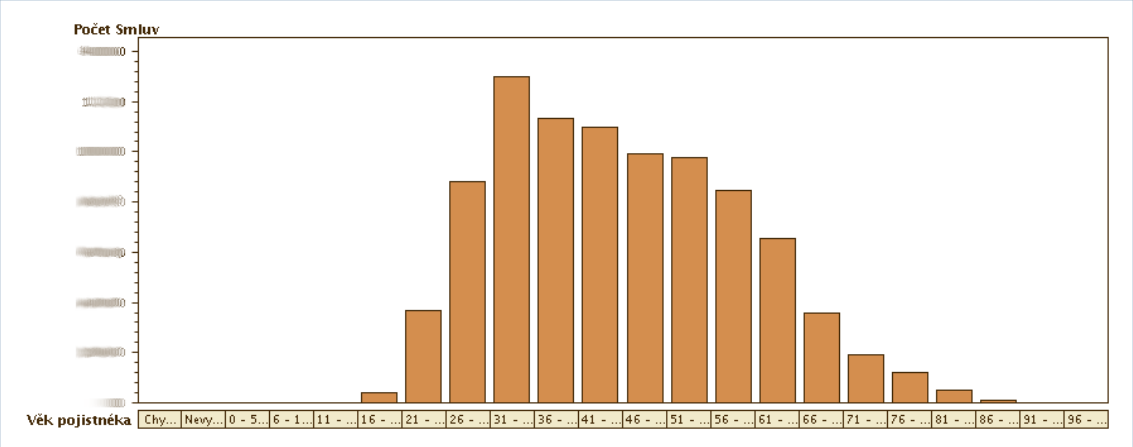
Počet obcí v ČR lze opět zjistit z údajů územně identifikačního registru adres. Z tabulky `obce` bylo zjištěno, že v ČR je 10 918 obcí s různým názvem.

PSČ

PSČ je dalším z údajů dostupných v územně identifikačním registru adres. Z tabulky `posta` bylo zjištěno, že v ČR existuje 3 520 unikátních poštovních směrovacích čísel.

Datum narození

Údaj o datu narození je poněkud problematické ohodnotit. Jedná se o údaj, který může, podle dat z datového skladu, nabývat hodnot od 1 do 140 let. Jako pomocný ukazatel pro návrh váhy tohoto atributu byl v datovém skladu vytvořen graf rozložení věkové struktury klientů pojišťovny (viz. Obrázek 5). Je vidět, že nejvíce klientů ČPP tvoří klienti ve věku 31 až 35 let, nicméně ostatní věkové skupiny nejsou od maximálních hodnot výrazně vzdáleny. Tento ukazatel tedy lze kvalifikovat jako vhodný pro rozhodovací kritérium.



Obrázek 5: Rozdělení počtu živých smluv občanských klientů podle věku pojistníka.

Návrh rozhodovací tabulky

Z předchozího textu vyplývají poměrně přesné údaje o statistickém výskytu jednotlivých hodnot jednotlivých atributů klienta:

Jméno	8 077
Příjmení	123 829
Věk	11 680
Ulice	27 413
Město	10 918
PSČ	3 520

Podle těchto údajů, a s přihlédnutím k výše uvedenému popisu jednotlivých atributů, byly stanoveny váhy kritérií pro model fuzzy unifikace. Vzhledem k subjektivní povaze hodnocení jednotlivých atributů bylo ke stanovení vah využito Saatyho metody.

Využití Saatyho metody

Saatyho metoda je využívána pro stanovení vah jednotlivých kritérií v modelu vícekriteriálního rozhodování v případě subjektivního hodnocení těchto kritérií. Metoda je založena na porovnávání všech možných kombinací dvou kritérií. Stupeň důležitosti jednoho kritéria vůči druhému je vyjadřován pomocí stupnice 1 – 9. Orientační hodnoty důležitosti je možné vyjádřit následujícím seznamem:

- 1 – shodná důležitost
- 3 – první kritérium je mírně důležitější než kritérium druhé
- 5 – první kritérium je značně důležitější než kritérium druhé
- 7 – první kritérium je silně důležitější než kritérium druhé
- 9 – první kritérium je absolutně důležitější než kritérium druhé

V rámci porovnávání důležitosti je samozřejmě možno využít i hodnoty 2, 4, 6, 8. Tyto hodnoty představují mezistupně. Hodnoty jsou následně umístěny do čtvercové matice k x k. Hodnota k je počet kritérií vícekriteriálního rozhodování. Hodnota h_{ij} obsahuje porovnání důležitosti těchto kritérií i-tého řádku vůči j-tému sloupci. Z tohoto pravidla vyplývají i další vlastnosti matice:

$$h_{ii} = 1$$

$$h_{ij} = 1 / h_{ji}$$

Váhy jednotlivých kritérií jsou definovány jako poměr hodnot geometrického průměru jednotlivých kritérií. Součet těchto vah je vždy roven jedné.

Tabulka 2 obsahuje navržené hodnoty pro úlohu unifikace klientů. Z výsledků je patrné, že nejdůležitějším atributem klienta je příjmení. Nicméně i jméno a ostatní atributy dosahují nezanedbatelné hodnoty.

	Jméno	Příjmení	Věk	Ulice	Město	PSČ	S^i	R^i	V^i
Jméno	1,00	0,20	5,00	4,00	3,00	5,00	60,00	2,27	0,25
Příjmení	5,00	1,00	7,00	3,00	3,00	5,00	1 575,00	4,36	0,47
Věk	0,20	0,14	1,00	0,20	0,33	1,00	0,00	0,29	0,03
Ulice	0,25	0,33	3,00	1,00	3,00	3,00	2,25	1,18	0,13
Město	0,33	0,33	3,00	0,33	1,00	3,00	0,33	0,80	0,09
PSČ	0,20	0,20	1,00	0,33	0,33	1,00	0,00	0,34	0,04

Tabulka 2: Stanovení vah kritérií pro model unifikace Saatyho metodou

Aspirační úrovně pro sloučení záznamů

Dalším krokem je nastavení hodnoty dosaženého hodnocení, při kterém lze považovat dva záznamy za shodné. Jak bylo uvedeno dříve, mezi kritéria nebyl z důvodu neúměrně velké preference, zařazen atribut rodné číslo. Tento atribut rozděluje záznamy na záznamy se správně vyplněným rodným číslem, které souhlasí s master záznamem, kde dochází pouze k „ověřujeme“, že nedošlo omylem k záměně rodného čísla. A na záznamy s nevyplněným, případně chybně vyplněným rodným číslem. U těchto dvou skupin záznamů jsou nastaveny rozdílné hodnoty aspiračního kritéria. V případě shody v rodném čísle byla tato hodnota stanovena na 0,5, v opačném případě na 0,8.

5 ZÁVĚR

Za použití metod operačního výzkumu, konkrétně modelů vícekriteriálního rozhodování, byl navržen a realizován jednoduchý proces unifikace klientů v datovém skladu. Proces byl vytvořen v prostředí datového skladu České podnikatelské pojišťovny, a.s., Viena Insurance Group. Testování proběhlo na desetiprocentním vzorku klientských dat pojišťovny. Technické údaje testu:

Počet klientských záznamů: 225 571

Počet záznamů občanských klientů: 205 083

Počet záznamů po aplikaci unifikace na 100% shodu: 204 923

Počet master záznamů: 183 965

Počet slave záznamů: 20 165

Počet unifikovaných záznamů pomocí rozhodovacího stromu: 714

Počet unifikovaných záznamů pomocí fuzzy unifikace: 190

Trvání testu: 36 hodin

Testem bylo prokázáno, že navrženou metodu unifikace je možno pro interní potřeby unifikace využít a unifikace je účinná. Pro reálné nasazení je třeba vyřešit některé technické požadavky. Jedná se zejména o optimalizaci celého kódu a zkrácení doby načítání. Zároveň je třeba dořešit otázku inkrementálního načítání. Je také třeba důkladně analyzovat výsledky unfikace a zvážit případnou změnu nastavení parametrů jednotlivých kroků unifikace. Z výsledků je například vidět menší účinnost fuzzy unifikace. Proces nastavování parametrů je však standardním a opakujícím se úkolem analytiků datového skladu.

6 Seznam Literatury

BROŽOVÁ, H. – HOUŠKA, M. – ŠUBRT, *T. Modely pro vícekriteriální rozhodování*. 1. vyd. Praha: Česká zemědělská univerzita v Praze, 2003. 172 s. ISBN 80-213-1019-7.

ČERNÝ, M. – GLÜCKAUFOVÁ, D. – TOMS, *M. Metody komplexního vyhodnocování variant*. 1. vyd. Praha: Academia, 1980. 227 s.

FIALA, P. – JABLONSKÝ, J. – MAŇAS, M. *Vícekriteriální rozhodování*. 1. vyd. Praha: Vysoká škola ekonomická v Praze, 1994. 316 s. ISBN: 80-7079-748-7.

FIALA, P. *Modely a metody rozhodování*. 1. vyd. Praha: Oeconomica, 2003. 292 s. ISBN 80-245-0622-X.

FOTR, J. – DĚDINA, J. – HRŮZOVÁ, H. *Manažerské rozhodování*. 3. vyd. Praha: Ekopress, 2003. 250 s. ISBN 80-86119-69-6.

W. H. INMON, *Building the Data Warehouse.* 4. ed. 2005. ISBN: 0-7645-9944-5

RŮŽIČKA, J. - ČÁBELKA, Š., *Přednáška k marketingové data quality.* Praha: 2007

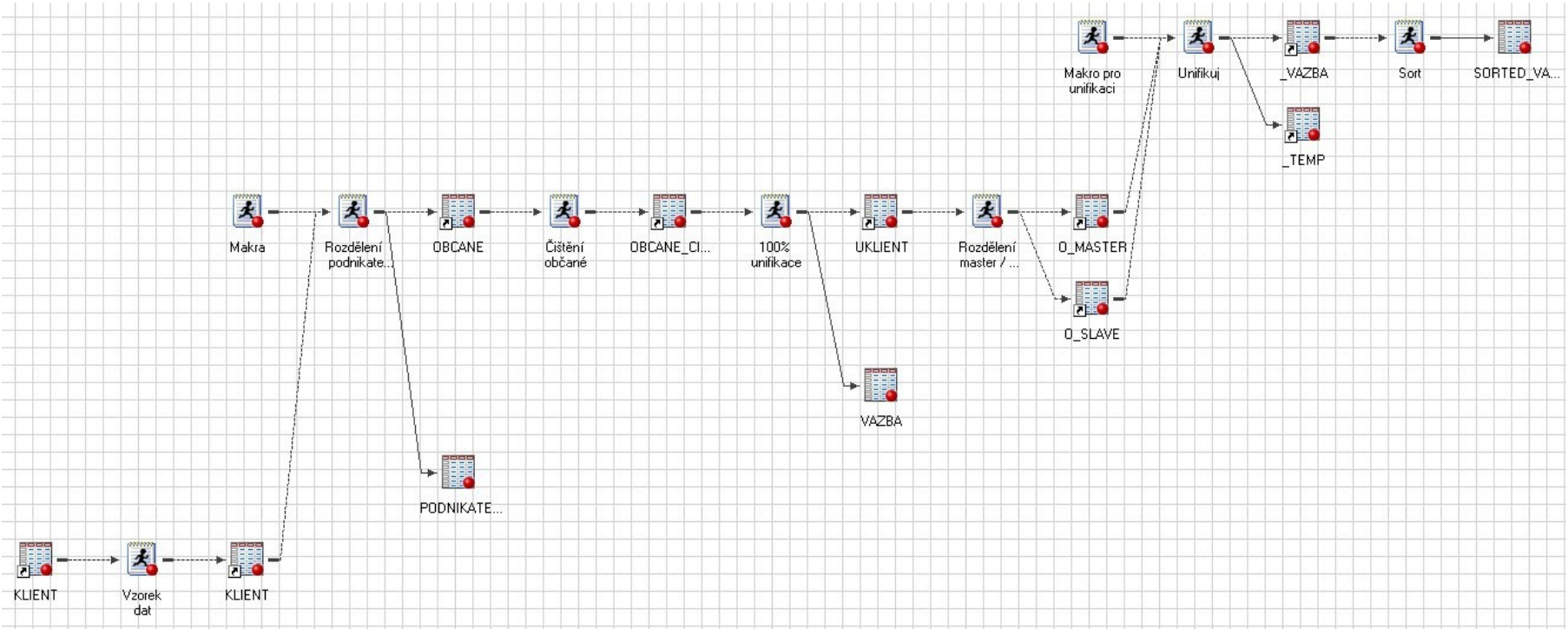
SAS 9.1 Documentation [online]. 19.10.2007 [cit. 14.01.2008]. URL: <http://support.sas.com/documentation/onlinedoc/91pdf/index.html>

Četnost jmen a příjmení – Ministrerstvo vnitra České republiky [online]. [cit. 28.02.2008]. URL: <http://www.mvcr.cz/clanek/cetnost-jmen-a- prijmeni.aspx>.

7 PŘÍLOHY

V rámci práce byl proveden test unifikace na reálných datech České podnikatelské pojišťovny, a.s., Viena Insurance Group. Kód byl vytvořen v datovém skladu pojišťovny na platformě SAS. Byl využit analytický nástroj Enterprise Guide v. 4.1.

7.1 Pohled na proces unifikace v Enterprise Guide



7.2 Zdrojový kód pro unifikaci

```
/*
 *Výběr 10% vzorku dat.
 */

data workspds.klient();
	set
		egdata.klient (
			rename = (
				obec = misto
			)
			where = (
				ranuni (5555) <.1
			)
		)
	;
run;

/*
 * Rozdělení záznamů na občanské a podnikatelské klienty
 */
data
	workspds.obcane (
		keep = part_pk typkl rc prijmeni jmeno uiradr ulice misto psc
		where = (
			typkl in ("FO")
		)
	)
	workspds.podnikatele (
		where = (
			typkl not in ("FO")
		)
	)
;
	set workspds.klient;
run;

/*
 * Vyčištění dat
 */
```

```
data workspds.obcane_cisty ();
      set workspds.obcane;
      %czdata_rc(rc=rc, dob=datnar, age=vek, sex=pohlavi, test=, tolerance=medium );
      %czdata_rc(rc=rc, dob=, age=, sex=, test=spravnost, tolerance=low );
      if(spravnost ne "") then rc = "";
run;

proc sort data=workspds.obcane_cisty out=workspds.obcane_cisty;
      by  rc jmeno uiradr ulice misto psc datnar vek pohlavi;
run;

/*
 * Unifikace na 100%
 */

data
      workspds.uklient (
            drop = upar_pk1 upar_pk2 upar_pk3
      )
      workspds.vazba (
            keep = part_pk upar_pk1 upar_pk2 upar_pk3
      )
;
      set workspds.obcane_cisty;
      by rc prijmeni jmeno uiradr ulice misto psc datnar vek pohlavi;

      attrib
            part_pk           label="Primární klíč partner"
            upar_pk1    label="Primární klíč unifikovaný partner 100%"
            upar_pk2    label="Primární klíč unifikovaný partner rozhodovací strom"
            upar_pk3    label="Primární klíč unifikovaný partner fuzzy"
      ;

      retain upart_pk1 0;
      upar_pk2 = 0;
      upar_pk3 = 0;

      if first.pohlavi then do;
            /* První výskyt daného klienta v tabulce */
```

```
            /* Zápis do tabulky unifikovaných klientů */
            output workspds.uklient;

            /* Zápis do tabulky vazeb */
            upar_pkl = part_pk;
            output workspds.vazba;
      end;
      else do;
            /* Další výskyty daného klienta v tabulce */
            /* Zápis do tabulky vazeb */
            output workspds.vazba;
      end;

run;

/*
 * Rozdělení záznamů na master a slave záznamy
 */

data
      work.o_master (
            where = (
                  spravnost eq ""
            )
            rename = (
                  part_pk = part_pk_m
                  rc = rc_m
                  prijmeni = prijmeni_m
                  jmeno = jmeno_m
                  uiradr = uiradr_m
                  ulice = ulice_m
                  misto = misto_m
                  psc = psc_m
                  datnar = datnar_m
                  vek = vek_m
                  pohlavi = pohlavi_m
            )
      )
      work.o_slave (
            where = (
                  spravnost ne ""
            )
            rename = (
```

```
                    part_pk = part_pk_s
                    rc = rc_s
                    prijmeni = prijmeni_s
                    jmeno = jmeno_s
                    uiradr = uiradr_s
                    ulice = ulice_s
                    misto = misto_s
                    psc = psc_s
                    datnar = datnar_s
                    vek = vek_s
                    pohlavi = pohlavi_s
              )
        )
;
        set workspds.uklient;
        by rc;

        if(compress(jmeno) eq "") then spravnost = "N";
        if(compress(prijmeni) eq "") then spravnost = "N";
        if(uiradr eq .) then spravnost = "N";
        if(compress(misto) eq "") then spravnost = "N";
        if(compress(psc) eq "") then spravnost = "N";

        if (rc ne "") then do;
              if(not first.rc) then do;
                    spravnost = "R";
              end;
        end;
run;

/*
 * Vytvoření dočasného kódu pro výběr cílových dat
 *
 * Vzhledem k obtížnosti a relativní nepřehlednosti SAS kódu
 * s použitím maker byla zvolena metoda vytvoření souboru,
 * který obsahuje spustitelný kód SAS. V následujícím kódu
 * dochází k vygenerování jiného SAS kódu a konci skriptu se
 * pak tento uměle vytvořený kód zavolá.
 * Struktura vytvářeného kódu je jednoduchá. Pro každý slave záznam
 * je vytvořena sekvence přákazů, ve které dojde k uložení atributů
 * tohoto záznamu do makroproměnných. Na konci každého záznamu je
 * pak voláno makro provádějící vlastní unifikaci
 */
```

```
data sasuser._vazba;
      attrib
            part_pk_s length=8
            part_pk_m length=8
            hodnota  length=8
            hodnotaf length=8
            /*score length=8*/

      ;
      stop;
run;

OPTIONS nonotes nosource;
proc printto log='/tmp/milan_run5';
run;

/*filename code temp; */
filename code "/tmp/milan";

data _null_;
      file code;
    set work.o_slave;

      k = '%let part_pk_s = ' || compress(part_pk_s) || ';';
    put k;

      aaa = '%let rc = "' || compress(rc_s) || '";';
    put aaa;

      b = '%let prijmeni = "' || compress(upcase(prijmeni_s)) || '";';
    put b;

      b = '%let jmeno = "' || compress(upcase(jmeno_s)) || '";';
    put b;

      c = '%let uiradr = ' || compress(uiradr_s) || ';';
    put c;

      d = '%let ulice = "' || compress(upcase(ulice_s)) || '";';
```

```
    put d;

      e = '%let misto = "' || compress(upcase(misto_s)) || '";';
    put e;

      f = '%let psc = "' || compress(upcase(psc_s)) || '";';
    put f;

      g = '%let datnar = ' || compress(datnar_s) || ';';
    put g;

      h = '%let vek = ' || compress(vek_s) || ';';
    put h;

      i = '%let pohlavi = "' || compress(pohlavi_s) || '";';
    put i;


    put'%unifikuj;';
run;

%include code; *spustí vlastní kód;

/*
 * Generování výstupní tabulky zdrojových dat pro zápis do
 * vazební tabulky
 */

PROC SORT DATA=sasuser._VAZBA
      OUT=WORK.Sorted_VAZBA(LABEL="Sorted WORK._VAZBA")
      ;
      BY part_pk_s hodnota hodnotaf;

RUN;

data work.prupis;
      set WORK.SORTED_VAZBA;
      by part_pk_s hodnota hodnotaf;

      if (first.part_pk_s) then output;
```

```
run;
```

7.3 Zdrojový kód makra

```
/*
 * Makro provádějící vlastní unifikaci klientů podle
 * rozhodovacího stromu a fuzzy unifikaci.
 * Z důvodu optimalizace jsou tyto dvě unifikace prováděny najednou
 */

%macro unifikuj;
       /*
        * Je vytvářena dočasná tabulka obsahují všechny dohledané a použitelné vazby
     * na master záznamy. Z těchto záznamů je následně vybírána
     * vazba s nejvyšším score
     */
       %put &part_pk_s;
       data work._temp (keep = part_pk_s part_pk_m hodnota hodnotaf);
              set work.o_master;
              hodnota=0;

              /* rozdělení podle rodného čísla na záznamy s a bez rodného čísla */
              if (length(&rc) > 1 and rc_m ne &rc) then do;
                     /* Slave záznam má vyplněno rodné číslo, čísla se ale neshodují */
                     part_pk_s = &part_pk_s;
                     hodnota = 0;
*                    output;
              end;
              else if (rc_m eq &rc) then do;
                     /* master a slave záznam se shodují v rodném čísle */
                     hodnota = 4;

                     if(compress(upcase(jmeno_m)) eq &jmeno and compress(upcase(prijmeni_m)) eq &prijmeni) then do;
                            /* shoda RČ, příjmení, jméno */
                            part_pk_s = &part_pk_s;
                            hodnota = hodnota + 6;
                            output;
                     end;
                     else if (uiradr_m eq &uiradr) then do;
                            /* shoda uir-adr */
```

```
                        part_pk_s = &part_pk_s;
                        hodnota = hodnota + 5;
                        output;
                  end;
                  else do;
                        /* shoda RČ, ale není shoda na nic dalšího */
                        part_pk_s = &part_pk_s;

                        /* pokus o Fuzzy unifikaci */
                        h_j = (length(compress(jmeno_m)) - complev(compress(jmeno_m), &jmeno,
'iln')) / length(compress(jmeno_m));
                        h_p = (length(compress(prijmeni_m)) - complev(compress(prijmeni_m), &prijmeni,
'iln')) / length(compress(prijmeni_m));
                        if(vek_m eq &vek) then h_v = 1;
                        else h_v = 0;
                        h_u = (length(compress(ulice_m)) - complev(compress(ulice_m), &ulice,
'iln')) / length(compress(ulice_m));
                        h_m = (length(compress(misto_m)) - complev(compress(misto_m), &misto,
'iln')) / length(compress(misto_m));
                        h_ps = (length(compress(psc_m)) - complev(compress(psc_m), &psc, 'iln')) /
length(compress(psc_m));

                        hodnota = 0;
                        hodnotaf = h_j*0.25 + h_p*0.47 + h_v*0.03 + h_u*0.13 + h_m*0.09 + h_ps*0.04;

                        if(hodnotaf > 0.4) then output;
                        *output;
                  end;
            end;
            else if (uiradr_m eq &uiradr) then do;
                  /* master a slave záznam se neshodují v rodném čísle, ale je shoda na UIR-ADR */
                  hodnota = 0;

                  if (compress(upcase(prijmeni_m)) eq &prijmeni) then do;
                        /* master a slave záznamy se nedhodují v rodném čísle, ale je dhosa na UIR-ADR
a příjmení */
                        part_pk_s = &part_pk_s;
                        hodnota = hodnota + 6;
                        output;
                  end;
```

```
                    else if(compress(upcase(jmeno_m)) eq &jmeno) then do;
                          /* master a slave záznamy se nedhodují v rodném čísle, ale je dhosa na UIR-ADR
a jménu */
                          part_pk_s = &part_pk_s;
                          hodnota = hodnota + 5;
                          output;
                    end;
                    else do;
                          /* master a slave záznamy se neshodují ani v RČ ani jinak; */
                          part_pk_s = &part_pk_s;

                          /* pokus o Fuzzy unifikaci */
                          h_j = (length(compress(jmeno_m)) - complev(compress(jmeno_m), &jmeno,
'iln')) / length(compress(jmeno_m));
                          h_p = psc;
                          if(vek_m eq &vek) then h_v = 1;
                          else h_v = 0;
                          h_u = (length(compress(ulice_m)) - complev(compress(ulice_m), &ulice,
'iln')) / length(compress(ulice_m));
                          h_m = (length(compress(misto_m)) - complev(compress(misto_m), &misto,
'iln')) / length(compress(misto_m));
                          h_ps = (length(compress(psc_m)) - complev(compress(psc_m), &psc, 'iln')) /
length(compress(psc_m));

                          hodnota = 0;
                          hodnotaf = h_j*0.25 + h_p*0.47 + h_v*0.03 + h_u*0.13 + h_m*0.09 + h_ps*0.04;
                          if (hodnotaf > 0.8) then output;
                          *output;
                    end;
              end;
              else do;
                          /* Není shoda RČ ani UIR-ADR */
                          part_pk_s = &part_pk_s;

                          /* pokus o Fuzzy unifikaci */
                          h_j = (length(compress(jmeno_m)) - complev(compress(jmeno_m), &jmeno,
'iln')) / length(compress(jmeno_m));
                          h_p = (length(compress(prijmeni_m)) - complev(compress(prijmeni_m), &prijmeni,
'iln')) / length(compress(prijmeni_m));
                          if(vek_m eq &vek) then h_v = 1;
```

```
                        else h_v = 0;
                        h_u = (length(compress(ulice_m)) - complev(compress(ulice_m), &ulice,
'iln')) / length(compress(ulice_m));
                        h_m = (length(compress(misto_m)) - complev(compress(misto_m), &misto,
'iln')) / length(compress(misto_m));
                        h_ps = (length(compress(psc_m)) - complev(compress(psc_m), &psc, 'iln')) /
length(compress(psc_m));

                        hodnota = 0;
                        hodnotaf = h_j*0.25 + h_p*0.47 + h_v*0.03 + h_u*0.13 + h_m*0.09 + h_ps*0.04;

                        if (hodnotaf > 0.8) then output;
                        *output;
            end;

      run;

      /* Vložení vazby s největším score do dočasné vazební tabulky */
      /* Pozor! Dočasná tabulka může obsahovat duplicity! */

      proc append base=sasuser._vazba data=work._temp;
      run;

      /* Uvolnění dočasné tabulky pro další použití */
      proc datasets lib=work nolist nowarn;
            delete _temp;
      run;


%mend unifikuj;
```