Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačního a znalostního inženýrství

Metriky informačních a datových struktur

DOKTORSKÁ DISERTAČNÍ PRÁCE

Doktorand:	Ing. Daniel Vodňanský
Školitel:	prof. RNDr. Jiří Ivánek, CSc.
Obor:	Aplikovaná informatika D-AI

Praha, 2021

Prohlášení

Prohlašuji, že doktorskou práci na téma „Metriky informačních a datových struktur“ jsem vypracoval samostatně. Použitou literaturu a podkladové materiály uvádím v přiloženém seznamu literatury.

V Praze dne 12. dubna 2021

……..

Podpis

Klíčová slova: Metriky, metadata, relační databáze, XML, JSON, RDF, ontologie, transformace, míra strukturovanosti, míra hierarchičnosti, míra informace, redundance, strukturalizace, hierarchizace, normalizace

Abstrakt: Práce popisuje univerzální rámec přístupu k heterogenním datům, informaci jimi reprezentované a typům jejich struktur.

Tento rámec umožňuje klasifikovat data na základě systému tří metrik – míry strukturovanosti, hierarchičnosti a informace a pomocí nich spočítat jejich vzájemnou kompatibilitu i kompatibilitu s různými datovými strukturami. Metriky společně umožňují vizualizaci problematiky prostřednictvím jejich chápání coby trojrozměrných souřadnic. To umožňuje matematicky klasifikovat typy datových transformací na základě jejich geometrického směru a délky a porovnávat tak různé alternativy při modelování datových schémat a návrhu datových transformací.

Keywords Metrics, metadata, relational database, XML, JSON, RDF, ontology, transformation, amount of structuredness, amount of hierarchicallity, amount of information, redundancy, structuralization, hierarchization normalization

Abstract: This thesis describes an universal framework, that approaches heterogenous data, the information it represents and their structure's types.

This framework enables data classification based on three metrics – structuredness, hierarchicallity and amount of information and use them to measure their mutual compatibility as well as compatibility to various data structures. Together, the metrics allow the problem to be visualized through their understanding as three-dimensional coordinates. This allows to classify data transformation types mathematically, based on their geometric direction and length and compare multiple alternatives while modelling data schemas and data transformation designing.

1 Obsah

2 Úvod

Tématem této práce je jednotný přístup k široké škále aktuálně používaných dat a typů jejich struktur s vazbou na informaci, jíž reprezentují. Práce se zabývá hledáním univerzální metody postavené na tvrdých metrikách, umožňujících ohodnotit data, jejich vazbu na typ úložiště (datové struktury), jejich vzájemnou podobnost a převoditelnost a změny stavu v čase.

Práce představuje trojici metrik – **míru strukturovanosti, míru hierarchičnosti a míru informace**, na jejichž osách lze data porovnávat. Díky tomu, že jde o trojici, je možné aplikovat srozumitelný geometricky orientovaný přístup k problematice a znázorňovat problémy v trojrozměrném prostoru. Kromě samotné definice metody výpočtu hodnot metrik navrhuje práce také metody strukturovaného uložení či průběžného ukládání naměřených hodnot a jejich vzájemné srovnání.

2.1 Vymezení tématu práce

2.1.1 Důvod výběru

Jednou z motivací výběru tohoto tématu je zájem o problém heterogenity hlavních ve světě používaných datových struktur a stále stejných opakujících se problémů v jejich vzájemné kompatibilitě (nejtypičtěji se jedná o relační databáze a formát XML, což studuje například (Krishnamurthy et al. 2001)). Téma této práce staví na tezi, že tato nekompatibilita není obvykle způsobena pouhou nekompatibilitou technologií nebo formátů úložišť, nýbrž často nekompatibilitou samotné daty reprezentované informace a celé ohraničené ontologie, představující řešenou oblast. Pohled na tuto problematiku se práce snaží sjednotit prostřednictvím studování sady použitelných informačně-datových metrik s navzájem porovnatelnými hodnotami.

Ačkoli ve světě vznikla a stále vzniká celá řada výzkumů na téma výpočtu informačních a datových charakteristik, v naprosté většině případů cílí pouze na jednotlivé charakteristiky nebo jednotlivé datové technologie (velmi často například relační databáze a formát XML). Někdy také ignorují podstatu jimi reprezentované informace, aniž by přistupovaly k heterogenní datové základně jako k více či méně integrovanému celku (konkrétní pří-

klady těchto výzkumů se nachází zejména v kapitole 6). Tato práce usiluje o syntézu jednotlivých přístupů jak k rovině dat a informace, tak i v přístupu k různým datovým technologiím, což umožní jejich vzájemné srovnání.

Výstupy této práce by měly například umožnit odpovídat na problémy znázorněné následujícími příklady.

> **Příklad 1**: je dána relační databáze. Cílem je jistou část jejích dat vhodným způsobem prezentovat na webu ve formátu HTML. Je tedy hledána vhodná vzájemně propojená podmnožina o jisté velikosti. Cílem je, aby výsledný HTML kód obsahoval co nejméně se opakujících sdělení (redundancí), aby byl ucelený, nemusel používat nepřehledné vnitřní odkazy na své části a maximálně využíval sémantické značky – obsahoval co nejméně volného textu. Jak ohodnotit, nakolik je konkrétní vybraná podmnožina dat schopna toto splnit?
>
> **Příklad 2**: jistý server produkuje periodicky jistý objem dat kombinující formáty XML a JSON, jejichž žádné schéma není dostupné. Je nutné zjistit, nakolik je potřeba tato data normalizovat a jaký je nejvhodnější způsob jejich ukládání – minimální velikost a minimální předpokládané vynaložené zdroje na jejich konverzi.

Jak ukazují oba příklady, za jistých podmínek lze konvertovat formáty dat a převádět jejich modely prakticky neomezeně. Ne vždy je to ovšem z mnoha důvodů zcela vhodné – krajním příkladem je uložení extrémně dlouhého textu do relační databáze, obsahující jednu tabulku s jedním řádkem i sloupcem nebo naopak konverze rozsáhlé strukturované relační databáze do prostého textu.

2.1.2 Vazba na výzkumný program pracoviště

Jedním ze stěžejních témat řešených Katedrou informačního a znalostního inženýrství Vysoké školy ekonomické v Praze jsou informační ontologie a problematika propojených dat na webu. S nimi spojený formát RDF (použitelný ke strukturované reprezentaci konkrétních faktů v prostředí webu) patří k jednomu z formátů analyzovaných v této práci v části 6.3. Spolu s vazbou na problematiku vizualizací, která je jedním ze zaměření této práce, je vhodné zmínit práci (Dudáš et al. 2018), byť tato vizualizační metoda cílí na úroveň jazyka OWL a tedy ontologických slovníků, nikoli konkrétních faktů.

Jedním z nejbližších řešených témat na katedře je pak problém řízení kvality dat a informací (Pejčoch 2014), jehož pohled na tématiku je ovšem orientovaný spíše na tvorbu podnikové manažerské metodiky pro samotné řízení kvality, nežli na jejich klasifikaci. Oproti tomu charakteristika metrik užitých touto prací není čistě kvalitativní a samotná datová kvalita nemá významný vztah k problematické zkoumání datové kompatibility a charakteristice datových transformací.

Další výraznější vazbou je výzkum webových technologií a sémantického webu, které jsou (díky komplexitě dat, která typicky využívají) jednou z inspirací této práce. Zde například (Hradil a Sklenák 2017) předvádějí základní pravidla pro implementaci rozhraní REST API – ta jsou z hlediska této práce zajímavá především užitím formátů XML a JSON.

Další vazbou katedrálního výzkumu je matematická logika a její jazyky (zejména logika deskripční), kterou tato práce klasifikuje coby jednu z nejpřesnějších reprezentací přímo na úrovni informace.

V souvislosti s ostatními blízkými pracovišti Fakulty informatiky a statistiky Vysoké školy ekonomické v Praze, lze využít informačně-datovou dimenzi v kontextu hospodářské organizace, tedy zejména podniku. Na úrovni **informace** jde o ohraničenou množinu informací, která je podniku či obdobné organizaci přístupná a potřebná pro její fungování – jde tedy o množinu konkrétních faktů. Na úrovni **dat** se poté jedná o data, která informaci reprezentují a ukládají chápaná coby aktivum – klíčovou součást majetku přinášející zisk. Kvalita této reprezentace informace těmito daty bývá také souhrnně označována jako datová kvalita, o které mluvila již sekce 2.1.2 a 2.2.

V tomto ohledu navrhuje (Närman et al. 2011) studium **datové kvality** a přesnosti pomocí konceptů **Enterprise Architecture**, přičemž uvádí „hlavní dimenze datové kvality“: „úplnost, konzistence, aktuálnost, relevance a přesnost“. Jak je však patrné, nejedná se o dimenze ve smyslu OLAP a jde spíše o konkrétní metriky datových jednotek. Podobně (Hoven 2003) chápe organizační datovou architekturu v souladu s řetězcem uvedeným v kapitole 4 coby propojení „datových zdrojů“ s potřebami byznysu. Informace a znalosti zde představují „zdroj“ pro samotná data. Toto propojení je složené ze standardů definujících strukturu (těmito standardy se následně zabývá (Hoven 2004)). Dále v kontextu Enterprise Architecture uvádí rámec **TOGAF** (The Open Group nedatováno) data coby jednu z architektonických domén vedle businessu, aplikace a technologie. **Datová architektura** je dle (Voříšek 2008, s. 133) jednou z dimenzí integrace podnikové informatiky a

z podnikového, potažmo i organizačního hlediska je definována jako „uspořádání datových zdrojů a informačních aktiv, kterými musí podnik disponovat, aby naplnil své definované cíle a potřeby“ (Voříšek a Pour 2012, s. 193).

Systematický přístup k problematice uspořádání datových zdrojů uvnitř organizace má velmi blízko k pojmu „**data governance**“ reprezentovaný institutem (Data Governance Institute nedatováno). Tento pojem poskytuje strukturovaný rámec definovaný coby „logická struktura pro klasifikování, organizování a komunikování komplexních aktivit zahrnujících tvorbu rozhodnutí o nich a přijímání opatření ohledně podnikových dat“. (Begg a Caira 2012) do pojmu data governance řadí disciplíny jako je zabezpečení, administrace a řízení kvality. Vztah k reprezentované informaci není v tomto přístupu systematicky řešen.

Především z hlediska datové kvality a obecně dat coby klíčového aktiva IT Governance dále rámec COBIT (ISACA 2019) uvádí požadavky na datovou kvalitu coby jeden ze vstupů (**APO14.07**) na operační úrovni požadavků manažerské praktiky **APO09.03** Definice a příprava servisních smluv. Dále je přímo uvedena praktika **APO14.04** – Definování strategie datové kvality. Zde do jisté míry s touto prací souvisí 5. aktivita – Vyhodnocení postupu, monitorování cílů a záměrů strategie datové kvality. Metriky, které tato práce přináší a především jejich využití ke zkoumání kompatibility dat a typů datových struktur lze pokládat za možnou součást evaluačního nástroje. Dále pak následující praktika **APO14.05** - Zavedení metodiky, procesy a nástroje pro profilování dat uvádí coby 5. aktivitu „Zajištění, aby byly výsledky centrálně uchovávány, systematicky sledovány a analyzovány s ohledem na statistiky a metriky“ – to odpovídá ukládacímu mechanismu metadatových skladů, které řeší tato práce v kapitole 8.

Standard DAMA-DMBOK (Cupoli et al. 2012), představující kolekci praktik pro disciplínu data managementu znázorňuje tuto oblast coby kruh, jehož středem je Data Governance. Ta je obalena oblastmi „datové architektury, modelování a návrhu, ukládání a operací, bezpečnosti, integrace a interoperability, dokument a obsah, reference a hlavní data, datové sklady a business intelligence (BI), metadata a datová kvalita“. S předpokládanou prací souvisí především oblast „ukládání“, která je ohodnocena trojicí navržených metrik, částečně i datových skladů a BI, jejichž principy jsou ovšem vztaženy na oblast metadat, tedy zjištěných hodnot metrik. Dále pak pochopitelně s prací do jisté míry souvisí již uvedená datová kvalita.

Celkově lze říct, že ačkoli práce **necílí na datovou kvalitu** jako celek, je možné metodu ohodnocování kompatibility dat a typů datových struktur chápat jako obecnou míru vhodnosti výběru typů struktur pro konkrétní data, která tím vypovídá o části kvality návrhu celé datové základny. Podobný je vztah i teorie informace v pojetí této práce ke konceptu Enterprise architecture a její datové podsložce – redundance dat a jejich kompatibilita se zvoleným typem datové struktury může sloužit jako kvalitativní ukazatel konkrétní datové architektury.

2.2 Stav řešení zkoumané problematiky

Jak ukazují zdroje uvedené především v kapitolách 4 a 6, v současné době existují izolované výzkumy zabývající se některými typy informačních i datových metrik. **Míra informace** je bezesporu desetiletí zavedený a zkoumaný pojem tvořící základ teorie informace, počínaje (Shannon 1948), potažmo její algoritmické pojetí (AIT) počínaje (Kolmogorov 1963), ovšem chybí její systematičtější aplikace v kontextu komplexních datových struktur. **Strukturovanost** bývá rozlišována na několika úrovních, ale jen výjimečně je objektem samotného výzkumu (potvrzuje i (Wang et al. 2012)), popřípadě nebývá chápána jako metrika, ale problém dat na jisté úrovni strukturovanosti (např. (Florescu 2005) řeší takto semistrukturovaná data). **Hierarchičnost** coby metrika v pojetí umožňujícím zkoumat kompatibilitu dat není patrně doposud řešena vůbec – už jen toto samotné slovo v české i anglické podobě (hierarchicallity) patrně doposud neexistuje. Ačkoli obdoby spojení dat a jejich hierarchií jsou používány velmi četně, nebylo nalezeno jejich použití v kontextu, který potřebuje tato práce, tedy zjištění, nakolik se datová struktura blíží struktuře stromu.

Právě četná existence izolovaných výzkumů obvykle na téma kompatibility nebo transformace dvojic datových formátů, datových modelů a dalších způsobů ukládání dat (např. RDBS a XML, jaký řeší například (Niewerth a Schwentick 2018), XML a JSON (Helland 2017)) bez kontextu obecné kompatibility jakýchkoli dat také dokládá, že téma zkoumané touto prací nebylo doposud na významné úrovni zkoumáno.

Dalším omezením řady potenciálně příbuzných výzkumů je velice častá ignorace informační úrovně a zkoumání pouze samotných dat, popřípadě zastřešení souladu dat s reprezentovanou informací pod do jisté míry odpovídající problém „datové kvality“. Samotná datová kvalita ovšem pro kontext této práce nepostačuje. Práce řeší datovou hete-

rogenitu a s tím související nesoulady datové reprezentace jako do jisté míry nevyhnutelný fakt, který je třeba co nejobjektivněji analyzovat po více osách i za účelem srovnávání. Nelze ho tedy vždy pouze hodnotit coby více či méně kvalitní, byť i zde jsou vždy některé hodnoty více žádoucí než jiné. Z kvalitativního hlediska uvádí např. (Floridi 2013) sadu metrik, konkrétně „přístupnost, přesnost, dostupnost, úplnost, aktuálnost, integrita, redundance, spolehlivost, včasnost, důvěryhodnost, použitelnost atd.“. Lze pokládat za zřejmé, že tyto metriky vyjma redundance, na níž stojí pojetí míry informace v této práci, nejsou pro zjištění stavu a vzájemné kompatibility dat jednotlivých typů struktur uvedených v kapitole 6 použitelné.

Jiné publikace naopak informační a datovou úroveň zcela zohledňují, jsou ovšem orientovány zejména manažersky a přímými výpočty charakteristik dat se nezabývají; to platí například pro DAMA-DMBOK (DAMA International 2017).

Celkově lze konstatovat, že tato problematika nebyla doposud přímo řešena pohledem blízkým pojetí této práce. Za současný stav lze tedy pokládat jednotlivé izolované výzkumy, jejichž aktuální stav je uveden v dotyčných kapitolách práce, zkoumajících jednotlivé struktury, metriky a jejich kombinace.

2.3 Cíle práce

Práce se zaměřuje na hledání systematického a matematicky orientovaného přístupu k heterogenitě dat a typů struktur, do kterých jsou uložena.

Práce má za svůj hlavní cíl návrh **výpočetní metody**, která dokáže komplexně analyzovat charakteristiky informačních a datových objektů za účelem srovnání jejich kompatibility a srovnání na časové ose, umožňující charakterizovat datovou transformaci. Datovým objektem se zpravidla myslí množina dat, reprezentující jednoznačně ohraničenou problematiku – množinu vzájemně propojených informací, případně i dat takto ohraničeným způsobem uložených (podrobněji definovány v částech 3.1, 4.1 a 0). Může se tedy jednat o množinu informací dostupných, vlastněných, spravovaných nebo využívaných jak technologicky, tak i skupinou zainteresovaných osob – může tedy jít o **firmu, úřad, aplikaci,** subjekt veřejné správy **nebo libovolný jiný typ lidské organizace** a jejich části. Výpočetní metoda musí být aplikovatelná na v současnosti typické (na trhu četně používané) datové struktury a data jim odpovídající a pokud možno být rozšířitelná na další formáty

a typy datových modelů. Dále musí umožňovat jejich srovnání – vzájemné a dále i na časové ose.

> **Prvním dílčím cílem** (DC1) je výpočetní metoda schopná charakterizovat, nakolik datový objekt obsahuje strukturovaná data a je tak vhodný k uložení strukturovaným způsobem. Hodnota je založena na zkoumání členitosti možného rozložení datového objektu na menší podobjekty.
>
> **Druhým dílčím cílem** (DC2) je výpočetní metoda schopná charakterizovat míru informace datového objektu. Zde je podstatné, nakolik datový objekt neobsahuje redundantní sdělení.
>
> **Třetím dílčím cílem** (DC3) je výpočetní metoda schopná charakterizovat, nakolik se grafová reprezentace složení datového objektu (je-li možné ji vytvořit) blíží stromu a je tak hierarchická – úměrně této hodnotě lze datový objekt ukládat ve formátu, který hierarchii vyžaduje.
>
> **Čtvrtým dílčím cílem** (DC4) je metoda vzájemného srovnání výsledných hodnot všech tří předchozích výpočetních metod, a to napříč datovým úložištěm i časem, umožňující grafickou vizualizaci.

Disciplínou, k níž se vztahuje tato práce, je Data Management, který se zabývá „vývojem, prováděním a dohledem nad plány, politikami, programy a postupy, které dodávají, kontrolují, chrání a zvyšují hodnotu datových a informačních aktiv po celou dobu jejich životnosti“ (DAMA International 2017, kap. 1). Pracovní rolí, na níž cílí praktické využití navržené metody, je primárně **datový kurátor** – pozice zabývající se „udržováním a správou metadat, nikoli databáze samotné“ (Foote 2019) a celkovým propojením oblastí informačních technologií a data science. Obecně metoda cílí na pracovní role navrhující procesy transformace dat, tedy v praxi nejčastěji vývojáře a správce datových úložišť.

Jedním z klíčových principů Data managementu plněných touto prací je řízení prostřednictvím metadat. V kontextu práce jde o hodnoty metrik v metadatovém skladu. Toto řízení přímo uvádí (DAMA International 2017, kap. 2.4) spolu se sadou souvisejících procesů: „tvorba, zpracování, použití, architektura, modelování, správa, řízení, řízení kvality dat, vývoj systémů, IT a business operace a analýzy“, přičemž výstupy této práce jsou relevantní k většině z nich vyjma „použití“ a „business operací“.

Indikátorem splnění hlavního cíle je praktická použitelnost metody ověřená ukázkovou analýzou, jejíž celková složitost umožňuje v případě demonstračních dat její zopakování

bez nutnosti automatizace výpočtu psaním výpočetních algoritmů. Výsledné hodnoty musí ukazovat, že postup jejich výpočtu naplňuje požadované vlastnosti každé ze tří metrik. Metoda musí být jednoznačně vyložitelná a použitelná na datech lišících se svým typem struktury i charakterem reprezentované informace a její výstup musí být schopen svými hodnotami tato data odlišit coby statický stav v časovém okamžiku i posun mezi stavy (datová transformace). Výstup dále musí být možné zaznamenat do navrženého úložiště umožňujícího jeho interpretaci i vizualizaci.

2.3.1 Výstupy

V souladu s dílčími cíli práce existují dva hlavní výstupy:

1. formalizovaná metoda klasifikace dat metadaty založenými na metrikách,
2. strukturované úložiště vypočtených hodnot, metoda jeho návrhu a způsobu naplnění hodnotami metrik a metoda jejich vizualizace.

Výstupy této práce se řídí několika následujícími požadavky.

Prvním požadavkem je přiměřená komplexita metody. Principy zde popsané, potažmo pak na nich založené aplikační výstupy a příklady použité k demonstraci, by měly být dostatečně jednoduché a srozumitelné. Mělo by být možné je aplikovat a interpretovat nejen vysoce specializovaným odborníkem na data, případně datovým kurátorem (které patrně lze očekávat více v korporátním sektoru), ale i IT pracovníkem v sektoru malých a středních firem (dále SME). Jak popisují (Begg a Caira 2012), tak rámce data governance sektor SME z převážné části opomíjí a zaměřují se na velké firmy a korporace. Přitom je nutné, aby metody byly aplikovatelné i na data v současnosti již existující. Toto kritérium je v souladu i s rozsahovým omezením práce. Na velmi rozsáhlá a komplexní data práce primárně necílí zejména z důvodu velmi vysoké pracnosti automatizování zde popsané metody (které vyžadují znalost struktury dat a reprezentované informace) pro tvorbu výpočetních algoritmů a tomu úměrné náročnosti procesu. Ovšem při použitelnosti metody na demonstračních datech malého rozsahu (tedy malé velikosti i komplexity) lze předpokládat (byť výrazně nákladnější) použitelnost i na datech podstatně větších.

Druhým požadavkem je škálovatelnost. Navržená řešení by měla být aplikovatelná na různě velké jednotlivé (v rámci jistého omezení popsaného prvním kritériem) informačně-datové objekty, představující například samotnou organizaci (podnik, úřad, aplikaci apod.) jako celek, její ohraničenou část, popřípadě i smysluplné sjednocení objektů

více souvisejících organizací a jejich částí. Řešení by mělo být možné dále rozšiřovat, a to zejména o nové metriky a dimenze nebo nové metody získávání hodnot stávajících.

Třetím požadavkem je obecnost. Výsledné řešení navržené touto prací by mělo být aplikovatelné na data o různém formátu (typu datové struktury), schématu (pokud existuje) i obsahu (včetně reprezentované informace).

2.3.2 Očekávané přínosy

V teoretické oblasti díky navrhovanému rámci umožňujícímu systematicky zkoumat stav i transformaci dat z hlediska jejich kompatibility vzniká základ možnosti dodat zpětnou vazbu **datovým transformačním procesům** na různé úrovni automatizace. Výpočet obecné kompatibility dat a typu datové struktury je nástrojem umožňujícím výběr nejvhodnější cílové struktury pro případ datové transformace. Zpětné ohodnocení transformace posouzením změny metrik naopak umožňuje odhalit případné nezamýšlené „vedlejší účinky“ transformace a tím srovnat její praktické provedení s původním předpokladem. Celkovým očekávaným přínosem v teoretické oblasti je unifikace pohledu na tento problém napříč spektrem nejtypičtějších zavedených datových struktur.

V praktické oblasti uveďme například proces hierarchizace dat, který může být systematicky řízen díky paralelnímu výpočtu míry informace a tedy i redundance. S její pomocí lze vybrat množinu dat pro hierarchizaci nejvhodnější (redundanci minimalizující). Může jít například o jisté položky relační databáze, které lze takto ohodnotit coby vhodné k automatickému generování HTML obsahu při minimálním vzniku redundance a maximálním zachování strukturovanosti. Vzhledem k neustálému rozvoji webových technologií pokládám tento příklad procesu za jeden z velmi klíčových. Proces strukturalizace lze optimalizovat obdobným způsobem – u strukturovaných dat lze předpokládat, že budou pokud možno normalizována, naopak jejich hierarchičnost může poklesnout.

Obdobně lze využít systém metrik i k ohodnocení opačného procesu transformace, kdy je z hierarchických, nestrukturovaných a silně redundantních dat modelována relační databáze.

Výstupy práce takto zaplňují prostor řady témat, o nichž mluví již část 2.2, který je zkoumán v současnosti spíše okrajově nebo příliš roztříštěně. V disciplíně Data managementu lze klasifikaci pomocí trojice metrik aplikovat především na

- doménu zajišťování kvality v rámci konceptu Data governance,
- zpětné ohodnocení datových modelů v rámci celkové datové architektury,
- detekci neobvyklých či podezřelých stavů v rámci datové administrace,
- ohodnocení datové transformace a kompatibility při návrhu a zpětné evaluaci procesu datové integrace,
- při získávání a zpracování dat v průběhu datové analýzy (metodu lze přímo pokládat za možnou součást datové analýzy),
- při správě metadat (naměřené hodnoty metrik jsou metadaty) a
- v kontextu principů datové kvality, vztah k nimž již popsala sekce 2.1.2.

V případě **budování nové datové** základny a to zejména ve stavu, kdy jsou již známy některé vnější datové zdroje a jejich podoba, popřípadě při úpravě stávajících datových základen může výstup této práce posloužit jako nástroj při **datové** integraci. Zjištěné hodnoty metrik mohou upozornit na redundance různého rozsahu, určit vhodnost výběru konkrétního datového formátu a pomoci navrhnout procesy plnění datové základny daty a jejich exportování za různými účely.

2.3.3 Předpoklady a omezení

Práce se nezabývá byznys aspekty ani architekturou samotných dat, na které je výpočet metrik aplikován. Jediným externím vstupem tohoto procesu je tak odborné pochopení dat a jejich významu dostatečné pro implementaci výpočetního a ukládacího mechanismu. Práce se přímo nezabývá problémem zkoumání tzv. datové kvality, ačkoli její výstup má k tématu datové kvality blízko a lze ho chápat jako její součást, jak uvedla již část 2.1.2.

Záměrem práce je nutnost **udržet celý koncept dostatečně univerzální** pro možné stavy dat. Práce tedy nemůže přijít s algoritmem, který by libovolná data zpracovával zcela automatizovaně – je zde nutné porozumění datům (podobně jako u procesu budování klasických metadatových skladů). Automatizace tohoto procesu by byla snahou o aplikaci metod umělé inteligence v této oblasti, což výrazně překračuje rozsahové možnosti práce. V práci tedy budou ukázány příklady zpracování nejtypičtějších typů struktur dat, ovšem proces budování metadatového skladu zůstává závislý na expertovi, který datům porozumí.

Dalším omezením práce je **neexistence dokonalé reprezentace informace** – metriky na informační úrovni proto nikdy nemohou být určovány zcela objektivním způsobem.

Dokonalá datová struktura, tedy struktura přímo podporující uložení informace o libovolném charakteru (napříč spektrem možných hodnot všech tří metrik) patrně doposud neexistuje. Metriky na úrovni informace jsou proto zkoumány zejména po teoretické stránce (část 4.2).

Posledním omezením je značná pracnost i výpočetní složitost budování tohoto řešení úměrná komplexitě zkoumané datové základny, kde je potřeba implementovat aplikační řešení ad-hoc. Tento problém je podobný jako v případě budování datového skladu (OLAP), ovšem navíc zde doposud neexistují podpůrné nástroje usnadňující tento postup a jejich tvorba výrazně překračuje rozsahové možnosti této práce. Proto je řešení demonstrováno zejména na datasetech menšího rozsahu, které zároveň umožňují větší názornost provedených postupů a výpočtů.

2.4 Použité metody

Hlavní metodou použitou v práci je **syntéza** vytříděných současných teoretických poznatků, prováděná v několika paralelních rovinách. První rovinou je členění problematiky na úroveň informace a dat. Druhou rovinou je členění dle jednotlivých metrik a třetí rovinou je členění dle typů informačně-datových struktur, které jsou postupně **kvalitativně analyzovány** a výsledek jejich analýzy je formalizován trojicí metrik.

Evaluační metodou výstupů práce je **experimentální** ověření jejich použitelnosti na případové studii, tedy sadě demonstračních dat a jejich vzájemných transformací.

Vytvořený rámec lze použít coby **kvantitativní analytickou metodu** založenou na **dedukci**, která je prakticky ověřena na demonstračních i reálných datech.

2.5 Návaznost kapitol

Struktura práce se inspiruje metodikou psaní odborných prací a článků od (Zachman 2009) a také obecnými a stanovenými univerzitními standardy zpracování disertačních a jiných kvalifikačních prací.

Kapitola 3 vymezuje multidimenzionální pojetí výpočtu hodnot metrik, zejména pak ve vztahu klasického OLAP přístupu a alternativního metadatového pojetí, které tato práce představuje. Dále vymezuje koncept rozloženého zkoumání množin dat coby objektů. Kapitola 4 představuje pojetí informace z hledisek aplikovaných v této práci s důrazem na

omezení možných přístupů. Dále zkoumá struktury na informační úrovni, přístup k mírám na úrovni informace a možné přístupy k co nejpřesnější reprezentaci informace (nejvíce odpovídající definici informace, nikoli dat). Kapitola 5 definuje metriky pro strukturovanost, hierarchičnost a míru informace na datové úrovni. Kapitola 6 uvádí v kontextu metrik v práci použité typy datových struktur. Dále se pak zabývá jejich srovnáním prostřednictvím systému metrik. Kapitola 7 zkoumá dynamiku hodnot metrik na časové ose a základní klasifikaci nejtypičtějších směrů transformace a jejich možných kombinací. Také se zabývá obecnou metodou výpočtu kompatibility dat a jejich struktur. Kapitola 8 popisuje obecný princip návrhu metadatového skladu umožňujícího výpočet na datech odpovídajícím typu datové struktury. Kapitola 9 uvádí ukázku možnosti zjištění hodnot metrik na informační úrovni. Kapitola 10 obsahuje demonstraci výpočtu hodnot metrik na datové úrovni, včetně demonstrace vybudování metadatového skladu, analýzy datové transformace a dlouhodobého monitorování datové základny.

2.6 Vymezení pojmů

Práce používá pojem **množina dat**, případně **množina informací** v kontextu, kdy nezáleží na jejich uspořádání a nejsou zohledňovány další vazby. Jejich setříděním za účelem aplikace výpočtu hodnot metrik a uložení do metadatového skladu (viz kapitola 8) vzniká **datový objekt**, reprezentující objekt informace, o nichž mluví část 4.1. Tyto a další pojmy práce specifikuje v příslušných kapitolách zabývajících se příslušnou teorií nebo navrženými principy. Zkratky uvádí část 14.5.

Dále práce zavádí pojmy odpovídající přímo třem **metrikám** (které definují sekce 4.2.1 - 4.2.3 a části 5.2 - 5.4) a šesti základním typům **transformací** (část 7.2). Dalším pojmem je **metadatový sklad** (kapitola 8), pomáhající rozčlenit datové objekty včetně jednoho **hlavního** datového objektu do jejich **podobjektů**, až nejmenších možných **elementárních objektů**. Dále umožňuje uložení vypočtených hodnot metrik.

Dále sekce 5.5.1 zavádí pojem **trojrozměrný metadatový prostor**, který rozvádí především pro účely možností vizualizace naměřených hodnot.

3 Metriky a dimenze

Hned v úvodu je nutné podotknout jeden z hlavních problémů, na který tato práce naráží – samotné **rozlišení dimenze a metriky**. Zde platí jisté odlišnosti oproti běžnému pojetí BI a OLAP principu, který tuto práci značně inspiroval a kterým je myšlen obecně známý široce definovaný princip ukládání hodnot do „tabulek faktů" odkazujících na „tabulky dimenzí". Zatímco v tomto běžném pojetí návrhu struktury datového skladu lze obvykle hodnoty přímo klasifikovat coby metriku, anebo dimenzi, v pojetí metadatového skladu navrženého touto prací může být řada hodnot využitelná oběma způsoby. Je například možné zkoumat strukturovanost dat na jednotlivých úrovních hierarchičnosti, či zkoumat hierarchičnost různě strukturovaných dat, což odpovídá mimo jiné klasickému principu kontingenční tabulky (které uvádí i (Voříšek a Pour 2012, s. 230)) a nástrojů pro její zpracování. V případě některých hodnot naopak je toto rozlišení zcela evidentní, například čas nebo prostá bitová velikost dat (v některých případech zaměněna za textovou délku).

3.1 Objekt a elementární objekt

Elementární objekty jsou zkoumány na úrovni informace (část 4.1) i dat (část 5.1). Jde o **nejmenší možné objekty**, pro které má smysl výpočet určených metrik a zařazení do dimenzí a které reprezentují jedno sdělení.

Na rozdíl od dimenze a metriky se jedná v tomto kontextu o pravděpodobně doposud neužívané a nezavedené pojmy. Vzájemně související elementární objekty jsou dále organizovány do nadřazených objektů, tedy jejich smysluplně propojených a jednotným způsobem pojmenovatelných množin. Objekty takto tvoří strom.

O **objektech** mluví tato práce ve chvíli, kdy jde o **ohraničené množiny informací nebo dat** (tvořící hierarchii s dalšími objekty na jedné z těchto dvou úrovní). Pakliže se naopak jedná o „objekty" reálného světa mimo tento přímý kontext, je použit pojem **entita**. Toto pojetí objektu se blíží pojmu „instance", která ovšem je zpravidla vázána na konkrétní třídu a práce jej tedy nepoužívá. Vzhledem k univerzálnímu pojetí objektů a jejich hierarchií nemusí taková třída existovat a takovýto pojem tedy nedává v kontextu práce smysl.

Hodnota daných metrik pro daný objekt v jistém časovém okamžiku charakterizuje stav objektu. Vzdálenost objektu v objektové hierarchii od svého nejnižšího elementárního podobjektu se nazývá **úroveň objektu**, přičemž elementární objekty mají úroveň rovnou nule.

Celý objekt, který je v jistý okamžik zkoumán a není chápán jako podobjekt některého dalšího, je nazýván **hlavní objekt**.

3.2 Metrika

Tato práce se snaží pojímat metriky způsobem dostatečně srozumitelným a jednoduchým, aby bylo možné je využít v sektoru **malých a středních firem** – SME (v souladu s konceptem řízení MBI (Voříšek nedatováno; Voříšek a Pour 2012)). Systém metrik má řadu potenciálních účelů využití. Může se jednat o manuální analýzu konkrétního datového objektu na bázi auditu, popřípadě pak systematické, automatizované a pravidelné sledování, které umožní generování pravidelných reportů, upozornění na neobvyklé výkyvy i přípravu dat pro metody dobývání znalostí.

Samotný pojem metrika je jen v IT odvětví široce používán v řadě kontextů a lze tak narazit na velký počet definic více či méně odpovídajících pojetí této práce. Například (Shah 2013) říká, že dobrá metrika musí být porovnatelná (její hodnoty musí jít srovnat navzájem, popřípadě i s hodnotami jiných metrik), srozumitelná, být poměrem nebo ohodnocením něčeho (což úzce souvisí s porovnatelností) a chování měnící (tedy využitelná a využívaná, mající vliv na rozhodovací procesy). Dále (Shah 2013) člení metriky na

- kvalitativní a kvantitativní (jak uvádí i (Voříšek a Pour 2012, s. 229)),
- zbytečné (neschopné měnit chování, měly by být ignorovány) a použitelné[1],
- průzkumné (je nutné je hledat a zkoumat) a vykazující (jednoznačné a okamžitě dostupné),
- zpětné (na základě minulosti) a dopředné (predikce pravděpodobné budoucnosti) a
- korelované (měnící hodnoty společně) a kauzální (mající prokazatelný vliv jedna na druhou).

Zde je nutno podotknout, že pojem „kauzalita" je v případě metrik přinejmenším sporný, neboť kauzalita se vztahuje na konkrétní jevy a ne přímo na metriky, byť ovlivňuje jejich hodnoty. Tato práce se zaměřuje čistě na kvantitativní metriky. Metriky orientované na samotnou kvalitu dat, o jakých mluví např. (Floridi 2013), nejsou z hlediska heterogenity a kompatibility datových formátů použitelné ke srovnávání.

Dále lze metriky z podstatně více podnikového úhlu pohledu dělit dle (Voříšek a Pour 2012, s. 229) na

- KPI (Key Performance Indicator), měřící jednotlivé činnosti a zdroje procesu a
- KGI (Key Goal Indicator), měřící celkovou kvalitu výstupu procesu.

Katalogu MBI (Voříšek, nedatováno) uvádí celkem 7 „Metrik datových zdrojů a jejich kvality" a to:

- I201: Objem spravovaných datových bází v GB
- I202: Objem opravných činností a činností při odhalování poškozených dat v člověkohodinách
- I203: Objem ztrát z nekvalitních dat v tis. Kč
- I204: Počet hodnot neodpovídajících slovníkovým
- I205: Počet nekonzistentních verzí informací o jedinečném objektu napříč systémy
- I206: Včasnost dodání dat uživatelům
- I207: Doba trvání zálohy datového zdroje

Tyto metriky do jisté míry pokrývají podobnou oblast jako metriky navrhované v této práci. Jsou definovány jako „sledovaná a měřená hodnota ukazatele pro potřeby řízení

[1] „vanity" a „actionable", obtížně přeložitelné

informatiky“, které se váží na 3 další složky: úlohy (v nichž budou užity), aplikace (jichž budou součástí) a dimenze (v nichž budou měřeny na principu BI). Konkrétně I201 má blízko k již zmíněné bitové velikosti, která ovšem neumí být omezena jen na velikosti kompletních datových bází a I204 a I205 se částečně blíží míře informace, resp. redundanci. Celkově ovšem jsou metriky v tomto výčtu pojaty spíše manažersky.

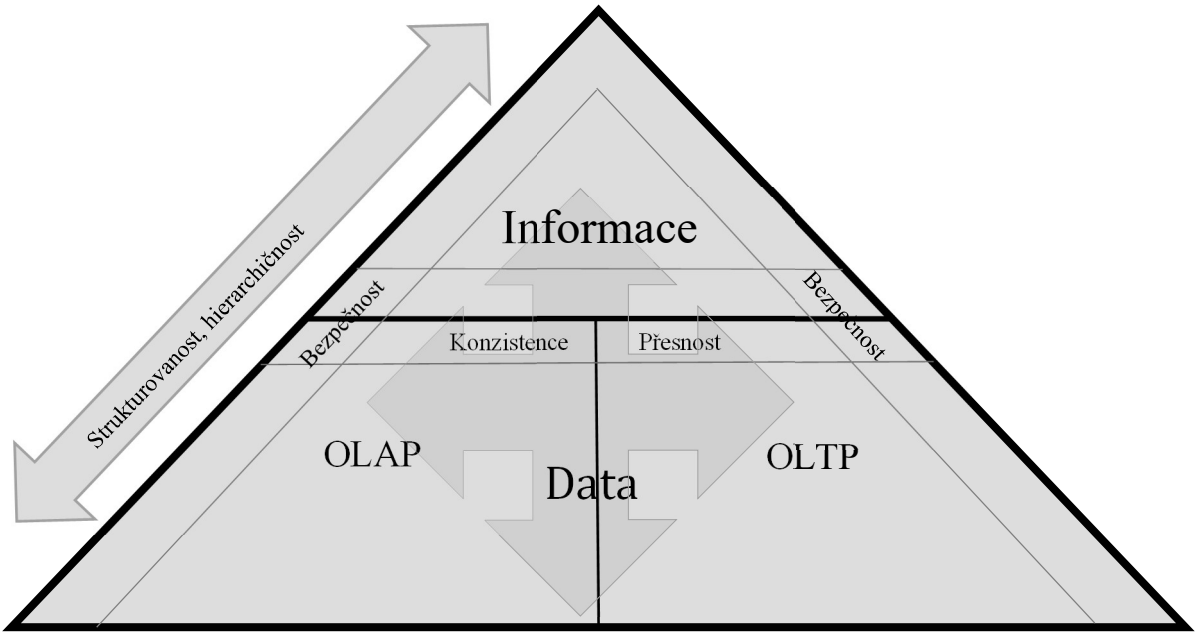
Dále je právě možné rozdělit metriky na **tvrdé** (exaktní, počítané matematicky, viz dále datové metriky v kapitole 5) a **měkké** (manažerské, založené více na subjektivním úsudku, viz předchozí odstavec a metriky informace v části 4.2). Tvrdá metrika je dle (Učeň, nedatováno) „objektivně měřitelný ukazatel“ převoditelný na finanční vyjádření (což je pro tuto práci poměrně nepodstatné), popřípadě pak indikátor, obsahující jasné spodní i horní meze. Oproti tomu měkké metriky mají dle (Business Performance Pty Ltd, nedatováno) výrazně subjektivnější charakter a bývají často zkoumány prostřednictvím většího počtu respondentů. Vzhledem k charakteru a tématu této práce a její vazbě na pracoviště jsou dále uvažovány **výhradně tvrdé metriky**.

Nejpodstatnějším aspektem pojetí metriky v této práci je vazba na její dimenze, které popisuje následující část a dále pak na objekt, na němž je měřena. Nejmenší možný, tedy **elementární** objekt informace nebo dat (podrobněji rozebíraný v částech 3.1 a 0) vyjadřuje typ objektu, který je nejmenším, pro nějž stále je výpočet metrik možný a dává smysl.

3.3 Dimenze

Pojem dimenze naráží v kontextu informace a dat na dvě možná pojetí.

Prvním, méně exaktním a spíše **manažerským pojetím** problematiky je dimenze coby úhel pohledu, jímž je možné na problematiku (zde především datovou základnu v organizaci) nahlížet, aby žádný aspekt nebyl opomenut. Tento přístup je inspirován metodikou MMDIS (Multidimensional Management and Development of Information System). Jejím hlavním principem je „paralelní řešení IS/ICT ze všech pohledů, které ovlivňují jeho efektivitu“ (Buchalcevová nedatováno), přičemž jednou z dimenzí je právě dimenze „data/informace“ (Voříšek, nedatováno). V souladu s tímto pojetím lze najít celou řadu dimenzí podnikových dat (zde coby „poddimenze“ dotyčné dimenze dle metodiky MMDIS), které se mohou navzájem protínat, přičemž konkrétní průsečík dvou nebo více dimenzí vyjadřuje možné téma (hledisko) k prozkoumání. Tento přístup znázorňuje Obrázek 1.



Obrázek 1 – dimenze podnikové informačně-datové základny

Tento přístup nicméně pro vysoký počet dimenzí a jejich nepříliš exaktní (měkký) charakter není pro tuto práci vhodný.

Druhým přístupem je pojetí známé z **metod BI**, kde dimenze v OLAP kostce vymezuje rozsah platnosti konkrétní hodnoty metriky. Tyto *multidimenzionální datové modely* (jak je nazývají již např. (Chaudhuri a Dayal 1997)) rozlišují dimenze od metrik přímo na úrovni tabulek relační databáze, mající nejčastěji strukturu „snowflake“, tedy sadu hierarchií dimenzionálních tabulek sbíhajících se do tzv. tabulky faktů, vyjadřující hodnoty konkrétních metrik. Tento přístup je v zásadě v souladu s definicí katalogu MBI, podle něhož se dimenzí „v MBI rozumí analytické hledisko pro identifikaci a hodnocení sledovaných ukazatelů a je tak součástí de facto každé metriky“ (MBI, nedatováno).

Tato práce se drží **druhého pojetí**, není-li explicitně uvedeno jinak.

4 Informační struktury

Tato kapitola s obdobným přístupem jako následující (kapitola 6) zkoumá struktury na úrovni informace. Hlavním účelem této kapitoly je vymezení konceptu metriky na informační vůči datové úrovni a znázornění faktu, že hodnota metriky se skládá jak z charakteristiky informace samotné, tak z charakteristiky dat, která jí reprezentují a ukládají.

Jedním ze základních prvků definujících úrovně poznání v této práci je obecně známý **řetězec pojmů**:

1. signál (ne vždy uváděn),
2. data,
3. informace,
4. znalost,
5. moudrost (ne vždy uváděna).

Ten lze sestavit syntézou podobných v literatuře zkoumaných (Bellinger et al. 2004; Tobin 1996; Liew 2007) či aplikovaných (Kanehisa et al. 2014) řetězců. Jejich vysvětlení je možné dvěma směry – ze strany příjemce sdělení (od bodu 1 do bodu 4, kdy stoupá příjemcovo porozumění) a ze strany zdroje (od bodu 4 do bodu 1, kdy je sdělení reprezentováno a kódováno). Bod 5 – pojem „moudrost“ přesahuje rámec této práce i kontext témat spadajících do oboru informatiky, a proto není definován ani dále zkoumán, ačkoli i pro něj oba směry dávají smysl.

V prvním směru lze odvodit, že signálem je myšlena změna fyzikální veličiny (napětí v kabelu, elektromagnetické či zvukové vlnění apod.). Tyto signály lze jistým způsobem dekódovat, čímž vznikají data o určitém typu (znak, číslice apod.). Pokud data vložíme do relací majících logický význam, vzniká informace – výrok s jistým způsobem ověřitelnou platností své hodnoty a v některých případech i převoditelný (normalizovatelný) do podoby **trojice** *subjekt – predikát – objekt* (např. *Jan Novák – se narodil – 1.1.1990* – toto chápání trojice je běžné ve formátu RDF). Z některých informací lze odvozovat nové informace, což odpovídá znalosti – například asociačnímu pravidlu (implikaci) s nejednoznačnou pravdivostí.

Ze strany informačního zdroje lze tento řetězec popsat opačným způsobem, tedy od znalosti, která je sdělitelná množinou informací. Informaci lze uložit prostřednictvím dat, která jsou následně zakódována do signálů.

Tato práce se zaměřuje na úroveň dat a informace; výpočty na úrovni signálu patří čistě do oboru fyziky, metriky znalostí patří odjakživa k základům disciplíny dobývání znalostí z dat a moudrost je vynechána z výše již uvedených důvodů.

4.1 Objekt a elementární objekt informace

Elementární objekt informace je v souladu s definicí z části 3.1 nejmenší možnou sdělitelnou informací. Pro účely této práce je za elementární informaci považována informace převoditelná do trojice *subjekt – predikát – objekt*, v pojetí známém již z formátu RDF.

Vzájemně související elementární objekty informace lze organizovat do objektů blízkých informačním ontologiím (jimiž je možné je reprezentovat), které ovšem nemusí být nijak explicitně formalizovány. Jedinou podmínkou chápání libovolné informační množiny coby objektu je **tedy schopnost ji zastřešit jednou samostatnou reálnou a pojmenovatelnou entitou**, s níž jako celek souvisí a je jí jednoznačně ohraničena (např. organizace, útvar, osoba nebo libovolný popisovaný objekt).

Podobný přístup používá v různorodém kontextu řada architektonických rámců. Například ArchiMate 3.0.1 (The Open Group 2017) používá pojem „**byznys objekt**“. Konkrétně uvádí, že „jazyk ArchiMate nedefinuje specifickou informační vrstvu, ovšem elementy jako byznys objekty a datové objekty jsou používány k reprezentaci informačních entit a logických datových komponent představujících byznys objekty“. Úzká vazba na aplikace a byznys je v pojetí této práce a kontextu klasifikace pomocí metrik výrazně nadbytečná, nicméně s konceptem prostého informačního objektu reprezentovaného datovými objekty je v souladu.

4.2 Metriky informace

Tato část se zabývá možnostmi zjištění charakteristik existující informace. Je nutné podotknout, že při jakémkoli výpočtu není reálně pracováno s informací samotnou, neboť ta je vždy (přinejmenším v případě stroje měření automatizujícího, u lidského mozku to lze pokládat za sporné) při svém zpracování **reprezentována jistými daty**, ať už více či méně kvalitně. Hlavní hranicí pro odlišení informačních struktur od struktur datových je tedy jejich primární zaměření na co nejpřesnější reprezentaci informace a nikoli přímé strojové použití. Metrika na úrovni informace tedy nemůže být na rozdíl od datových metrik přímo spočítána a jde tak o **měkkou metriku**.

4.2.1 Strukturovanost informace

Strukturovaností je dle (Boehm, B. W. et al. 1973) myšlen „stupeň, do jaké míry systém či jeho komponenta disponuje jistým vzorem svých vzájemně závislých částí“.

Na úrovni informace je problematické ptát se na strukturovanost informace samotné, neboť tato otázka míří často ve skutečnosti na strukturovanost konkrétního informačního sdělení, což lze chápat spíše jako datovou reprezentaci informace, než informaci samotnou. Pokud navíc informace (abstrahujeme-li od jakékoli vágnosti či neurčitosti) je normalizována do tvaru subjekt – predikát – objekt (A-Box), činí ji to samu o sobě zcela strukturovanou – nelze ovšem říct, zda je toto vždy možné. Totéž platí pro terminologický popis ontologie (T-box).

V odborné literatuře nicméně je pojem strukturovanost[2] studován v několika málo případech i na úrovni informace. V tomto ohledu má tomuto pojetí blízko (Laue a Mendling 2010), chápající strukturovanost (zde nazvanou „stupeň strukturovanosti“) coby hlavní metriku správnosti procesního modelování. Toto pojetí strukturovanosti je ovšem zaměřeno na procesy, nikoli konkrétní stav reality, a proto není tato teorie ani její výpočetní model pro tuto práci použitelná. Praktickou absenci výzkumu strukturovanosti informace potvrzuje i (Wang et al. 2012), který zmiňuje možnost jejího výpočtu pomocí střední míry informace.

Po praktické stránce je nutné připustit, že v řadě případů informace, ačkoli její podstata sama o sobě je strukturovaná, nemusí být v strukturované podobě dostupná, popřípadě strukturovaným způsobem sdělitelná. V řadě takovýchto případů by strukturalizace znamenala reálnou informační ztrátu, je-li informace vázána nejen na sdělení samotná, ale i na jejich (v tomto případě obvykle textovou) reprezentaci. Tato možnost informační ztráty se týká zejména textů určených pro člověka (včetně této práce) a uspořádaných pro snadné pochopení a zpracování lidským mozkem, ale také například (v krajním případě) díla umělecká. Informace v těchto textech se nemusí zakládat pouze na sémantickém významu jednotlivých výroků, ale i na jejich uspořádání nebo emočním zabarvení, která utváří jakousi síť vzájemných fuzzy relací jednotlivých sdělení. Celkový význam tak může být natolik nejednoznačným a subjektivním, že je prakticky nemožné ho systematicky transformovat do strukturované podoby.

[2] v angličtině používán pojem „structuredness“

Po zohlednění tohoto omezení je vhodné stanovit následující interní definici: **informace je strukturovaná právě do takové míry, jako je strukturovaná její nejstrukturovanější možná přesná datová reprezentace** (jejíž stanovení řeší sekce 5.2 na straně 43). V případě, kdy nejstrukturovanější možnou reprezentací je prostý text pak tedy lze mluvit o nestrukturované informaci. Problém určení míry strukturovanosti již na úrovni informace znázorňuje **Příklad 3**.

> **Příklad 3**: Modelujeme úložiště pro informace, jejichž vznik předpokládáme v budoucnu a momentálně existuje pouze jejich popis. Byly nalezeny tři oblasti, jichž se dotýká. První jsou účetní záznamy, druhou sbírka kuchyňských receptů a třetí popis historie firmy. Pokud takovéto informace v konkrétní podobě získáme, nakolik můžeme rozhodnout, zda bude možné je uložit strukturovaným způsobem?

4.2.2 Hierarchičnost informace

Pojem hierarchie je zpravidla chápán jako struktura mající stromový charakter, tedy skládající se z jedné „kořenové“ entity (pokud mluvíme o hierarchiích obecně, kde dotyčnou entitou může být cokoli), větvící se do entit dalších až po koncové, někdy také nazývané „listy stromu“. Dle (Musca et al. 2011) je struktura hierarchická, pokud jsou „jednotky na nižší úrovni součástí jednotek na úrovni vyšší“.

Pro ohodnocení hierarchičnosti je nutné informační strukturu zanést do orientovaného grafu, což do značné míry vyžaduje vyšší míru strukturovanosti a úzce tak souvisí s předchozí metodikou v sekci 4.2.1. V tomto orientovaném grafu platí, že množina uzlů je hierarchická, pakliže každý takovýto uzel odkazuje maximálně na jeden další uzel (který je také součástí této množiny) a sám je odkazován libovolným počtem dalších jiných uzlů. Matematicky je možné tento unární predikát hierarchičnosti $H(a)$ formalizovat následovně:

$$\forall b\colon R(a,b) \Rightarrow \left(\neg\exists c\colon c \neq b \wedge R(a,c)\right),$$

Vzorec 1 – definice hierarchické vlastnosti H uzlu a

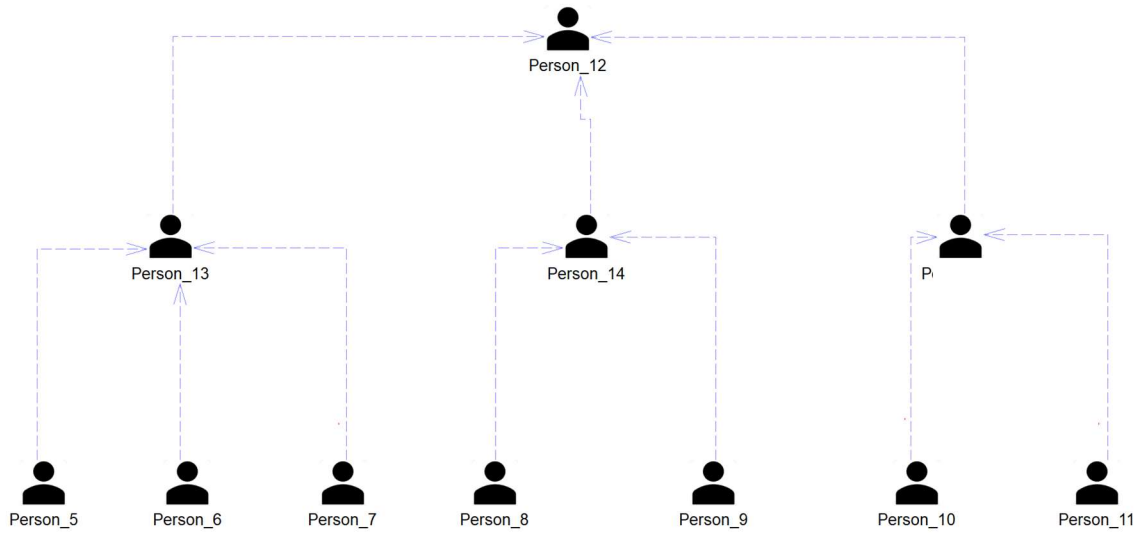
kde a, b a c jsou uzly grafu a R je binární relace dvou uzlů, kde první odkazuje na druhý.

Definice hierarchické struktury sama o sobě je četně řešené téma spadající do oblasti teorie grafů, ovšem tato práce neřeší pouze problém, zda struktura splňuje definici, nýbrž problém, jakým způsobem zjistit, **nakolik k jejímu splnění dochází**.

Podstatným problémem hierarchičnosti případná možnost existence cyklu, který definice (Vzorec 1) nevylučuje. I zde platí, že struktura musí být transformovatelná do ryze hierarchické podoby, která jako taková cyklus vylučuje a má-li být struktura hierarchická, pak je tedy cyklus možný pouze na úrovni T-Boxu (např. kusovník, tedy cyklus na úrovni schématu – rekurzivní relace), nikoli ovšem na úrovni A-Boxu – cyklus v samotné informaci (potažmo datech) tuto transformaci vylučuje – není možné vytvořit nekonečný hierarchický datový soubor.

Informace hierarchického charakteru, také nazývané stromy jsou dle (Eberhart et al. nedatováno) „fundamentální strukturou pro organizování znalostí“. Lze se tedy domnívat (ačkoli není možné to potvrdit), že jsou i jedním z důležitých typů struktur, s nimiž v jeden okamžik pracuje (na vědomé úrovni) lidský mozek. Je pravděpodobné, že mu dodává pocit přehlednosti a takto reprezentovaná informace je tak snadno zpracovatelná i dále sdělitelná. Je tedy pochopitelné, že na hierarchickou strukturu narážíme při každé strukturované (nejen elektronické) komunikaci – v dokumentech, organizačních schématech, jednotlivých webových stránkách, formulářích, prezentacích i při ostatních zpracováních vjemů světa, byť dotyčné realitě přesněji vyhovují například grafově orientované struktury.

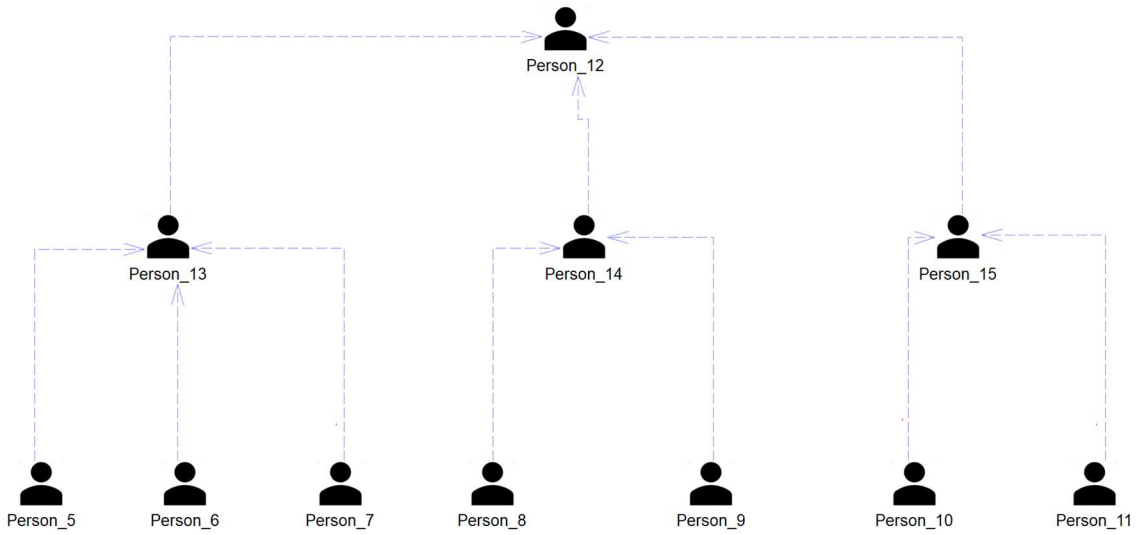
Příkladem hierarchické struktury je i tato disertační práce, nejsou-li brány v potaz vnitřní křížové odkazy, které jsou indikátorem skutečnosti, že realita touto prací popisovaná hierarchickou strukturu nemá.



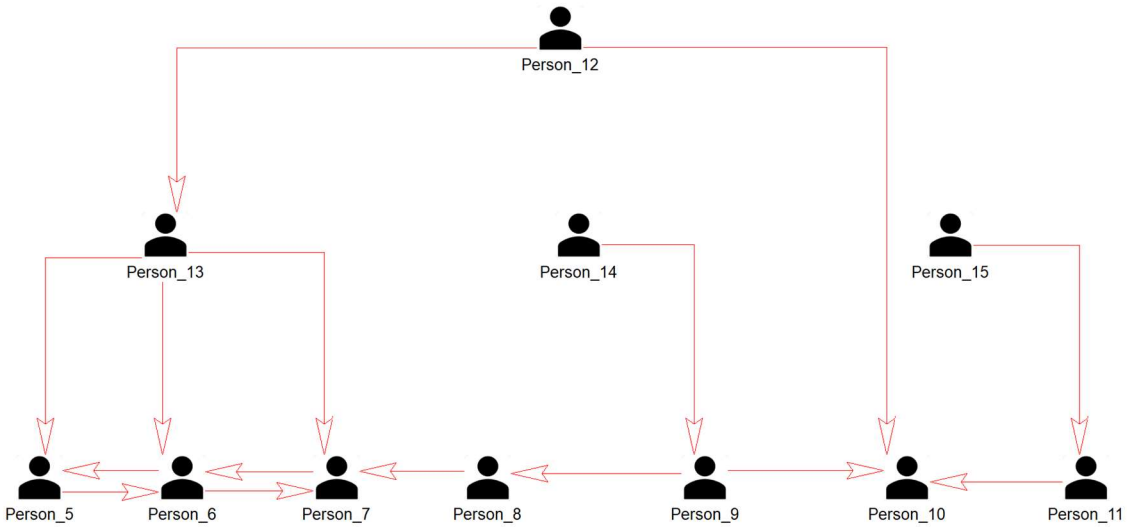
Obrázek 2 – lidé v organizaci z hlediska organizační struktury

To ovšem neznamená, že podstata informace reálného světa musí být většinově hierarchická; hierarchie zpravidla reprezentuje jistý úhel pohledu člověka na danou problematiku.

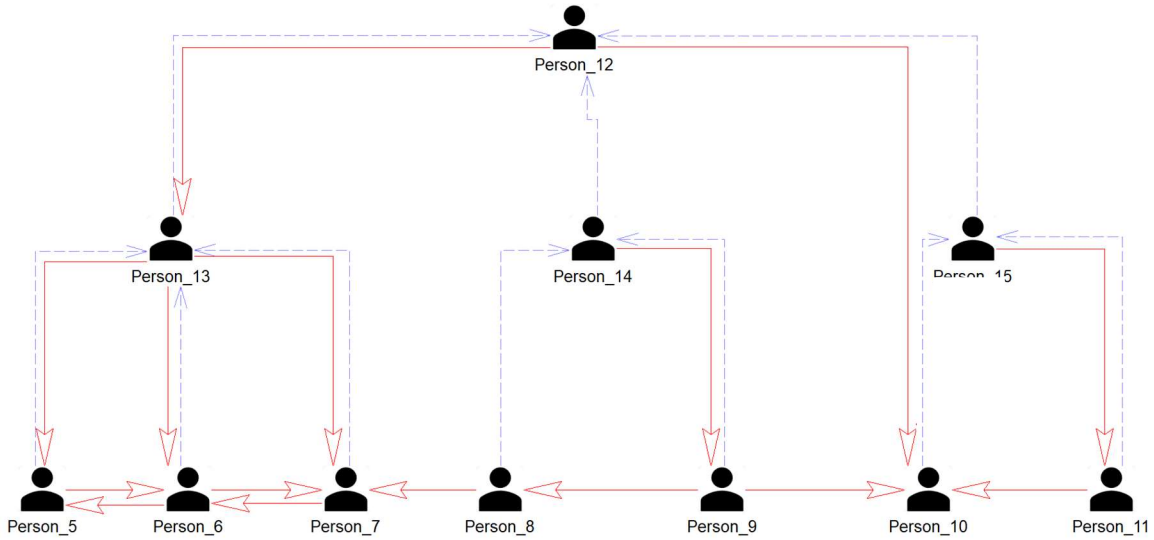
Příklad 4: pohled na organizační schéma podniku bezpochyby obvykle zprostředkovává hierarchickou informaci, zohledňuje-li výhradně organizační aspekty dotyčné organizace (ty jsou úhlem pohledu na množinu osob), jak ilustruje



Obrázek 2. Úhel pohledu na tytéž entity (osoby) z hlediska mezilidských vztahů nebo komunikačních proudů utváří strukturu zcela odlišnou, mající zcela odlišnou (v tomto případě patrně nižší) hierarchičnost, jak ukazuje Obrázek 3. Je-li pak realita zkoumána komplexním pohledem, zachycujícím mnohá možná hlediska reality a tím vícenásobné úhly pohledu, jsou vícenásobné struktury propojeny do jedné, jak ukazuje na příkladu Obrázek 4.



Obrázek 3 – lidé v organizaci z hlediska komunikačních proudů



Obrázek 4 – sjednocení obou hledisek

Podstatné je, že se stále jedná o stejné entity, viděné z různých úhlů pohledu. I ve zcela nehierarchické struktuře lze často najít hierarchickou substrukturu (v případě diagramu na Obrázek 3 jde o entity s čísly 10, 11, 12 a 15 – jsou-li chápány jako samostatná oddělená struktura).

Jednotlivé lidské úhly pohledu na výseč reality světa tedy k hierarchickému vidění inklinují, což se odráží i na modelování reality, jak ukazuje i výsledek zkoumání sady ontologických slovníků v (Vodňanský a Zamazal 2016). Toto zkoumání ukazuje velmi výraznou

převahu silně hierarchických struktur v ontologiích. Převaha hierarchičnosti může vést k mylné (byť často nevědomé) představě, že celková podstata ontologie světa je hierarchická a že absolutní hierarchičnost jakékoli modelované ontologie je v každém případě nutností. Z toho může vyplynout i chybné nebo nepřesné informační a datové modelování. Hierarchická vlastnost informační struktury je tedy klíčová pro možnost prezentovat informaci hierarchickým způsobem popsaným v předchozím odstavci, z hlediska IT například na jedné webové stránce (ačkoli web jako globální celek samozřejmě hierarchický není).

Tento problém lze pokládat za jeden z nejméně doposud zkoumaných. V literatuře se hierarchie vyskytují nejčastěji jen jako okrajové a nevymezené definice v mnoha různých kontextech. Například (Zach et al. 2014) mluví o méně hierarchické struktuře podniků v sektoru SME nebo (Shepelev 2011) se zabývá identifikací hierarchických struktur prostřednictvím neuronových sítí (perceptronů), paradoxně ovšem bez jejich přímé definice.

Ohledně již zmíněného problému stanovení hierarchičnosti při nižší úrovni strukturovanosti lze v souladu s principem, jaký uvádí již Vzorec 1 stanovit, že **nestrukturovaná informace je vždy hierarchická.** Neobsahuje totiž žádné vnitřní explicitní reference a nemůže tak porušit definici hierarchičnosti – fakticky se jedná o orientovaný graf obsahující pouze jeden neodkazující uzel. Hierarchičnost informace je pak **podíl velikosti dat splňujících definici hierarchičnosti na všech datech, která dotyčnou informační oblast reprezentují nejpřesnějším možným způsobem**.

4.2.3 Míra informace

Jednou z nejvýznamnějších teorií v oblasti dat, informace a jejich vzájemného vztahu je princip střední míry informace, též známé jako entropie, kterou představil (Shannon 1948). Míra informace obsažené v datech je zde chápána coby hodnota přímo úměrná nepravděpodobnosti příjmu jistého sdělení – přesně je mluveno o informačním zdroji produkujícím n různých typů zpráv x_i o pravděpodobnosti produkování $P(x_i)$. Nejvyšší míru informace má tak informační zdroj o největší možné nepředvídatelnosti svých sdělení, jehož typy zpráv mají stejnou pravděpodobnost $P(x) = \frac{1}{n}$. Výsledná hodnota H (Vzorec 2 – Shannonova střední míra informace) říká, nakolik přijetím zpráv klesá neurčitost na straně příjemce. Základ logaritmu b bývá obvykle stanoven jako 2, neboť v takovém případě entropie určuje minimální počet bitů nutných k zakódování dotyčných dat.

$$H = -\sum_i p_i \cdot \log_b p_i$$

Vzorec 2 – Shannonova střední míra informace (Shannon 1948)

Tento koncept ovšem sám o sobě není pro řadu struktur zkoumaných touto prací použitelný. Důvodem jsou zejména jeho následující omezení:

1. **Jednorozměrná data**. Pravděpodobnost (či její náhrada obdobnou hodnotou) je použitelná pouze na jednorozměrné sekvenci. Nelze tedy jednoznačně určit, jaký přístup zvolit v případě víceúrovňových hierarchií (stromů). Výše zmíněný informační kanál může například rozlišovat nejen jednotlivé zprávy, ale i podzprávy mající vlastní pravděpodobnostní hodnotu. Ještě viditelněji se toto omezení vztahuje na síťové struktury provázané libovolnou množinou relací.
2. **Nerozlišené pořadí**. Shannonův přístup zkoumá výhradně množinu typů zpráv jako celek, ačkoli je zřejmé, že míra neurčitosti může být ovlivněna i pořadím produkování zpráv v časové sekvenci, případně uspořádaností dat na jiné než časové ose. Toto pořadí tedy samo o sobě (každá permutace jednotlivých hodnot) nese určitou informaci. V případě tohoto problému je koncept Shannonovy míry rozšiřitelný o tzv. podmíněnou entropii, zavádějící popis náhodné veličiny o jedné pravděpodobnostní funkci $p(x)$ pomocí jiné funkce $q(x)$, což na základě vzdálenosti obou funkcí umožňuje vypočítat relativní entropii $D\big(p(x)\|q(x)\big)$ o hodnotě podmíněné vzájemné informace.
3. **Částečná podobnost**. Shannonův přístup počítá pravděpodobnost každé unikátní hodnoty, ty si ovšem mohou být do jisté míry podobné (např. různě komprimované obrázky se stejným obsahem) a tím reprezentovat i (přinejmenším pro člověka) identickou informaci. Možnostmi rozšíření entropie o fuzzy koncept, který toto řeší, se zabývají (Kosko 1986; Galar et al. 2010), ovšem do jisté míry je řešena konceptem algoritmické míry informace, popsané dále v této části.
4. **Strukturovaná data**. Toto omezení silně souvisí již s prvním bodem. Data musí být rozložitelná na smysluplnou sekvenci hodnot, což různé vágní a nejednoznačné relace dat a nejednoznačná ohraničitelnost sdělení obvykle neumožňuje. V tomto ohledu se například (Guerrero 2012) zabývá výpočtem entropie z textu a to srovnáním přístupů na základě hodnot coby slov a n-písmenných sekvencí, které oba poskytují obdobné výsledky.

Jednotlivá omezení jsou tedy poměrně často řešena současným výzkumem, jak ukazuje tento jejich výčet. Žádné z těchto řešení ovšem není natolik komplexní, aby řešilo všechna tato omezení, a to ideálně jednotnou metodou.

Problém vícerozměrných a ne zcela strukturovaných hodnot prakticky řeší především výzkum entropie na formátu XML, o níž dále mluví část 6.2. Obecně se většina omezení odvíjí od problému, jakou a jak velkou substrukturu chápat coby zprávu za účelem výpočtu pravděpodobnosti dle Shannonova vzorce. Na úrovni informace by mělo jít o nejmenší substrukturu reprezentující informaci.

Celkově lze říct, že ačkoli je Shannonovo pojetí logicky jedním ze základních přístupů k výpočtu míry informace, zejména uvedená omezení ho činí velmi neflexibilním pro praktické použití na silně heterogenních datech. Na druhou stranu ovšem samotná filosofie „předvídatelnosti“ sdělení je kvalitní inspirací k dalším přístupům, jako AIT (následující odstavec). Omezení jsou částečně odstranitelná řadou dalších alternativních přístupů, mezi které patří (Thadani a McKeown 2008), řešící identifikaci redundantní informační reprezentace v textu přirozeného jazyka. Tento přístup je práci velmi blízký, ovšem není dostatečně široký pro použití na textových řetězcích neodpovídajících přirozenému jazyku ani jiných než textových datech.

Druhým často uváděným přístupem k výpočtu informace v datech je tzv. algoritmická teorie informace (AIT). Výchozím teoretickým konceptem je tzv. Kolmogorova komplexita (Kolmogorov 1963). Dle (Hutter 2007) se „zabývá vztahy mezi výpočetní metodou, informací a nahodilostí“ a měří tak zejména právě komplexitu příslušných dat. Ta je vyjádřena jako délka nejkratšího popisu (obvykle je mluveno o programu) vyjadřujícího konkrétní textový řetězec, tedy například řetězec „01010101010101010101010101010“ může být takto zjednodušeně řečeno popsán jako „16 opakování řetězce ‘01’“. Namísto délky uvádí (Grünwald a Vitányi 2008) „minimální počet bitů potřebných k popisu pozorování“, což platí i pro Shannonův přístup.

Proces tohoto zkráceného zápisu dat je nazýván datovou kompresí a primárně slouží zejména k archivaci a zmenšení velikosti dat, nežli k samotnému výpočtu. Jeho výsledek je reprezentován jednak velikosti komprimované struktury a také jejím kompresním poměrem vůči velikosti struktury původní. Vzhledem k existenci celé řady algoritmů ((Hutter 2007) uvádí primárně kompresory Lempel-Ziv, bzip2 a PPMZ) tento přístup překonává veškerá uvedená omezení Shannonova přístupu. Na druhou stranu nelze nikdy za-

ručit, že výsledná zkomprimovaná velikost bude nejmenší možnou. Na toto téma pravděpodobně neexistuje relevantní vědecká studie a její vypracování by nebylo v dokonalé podobě ani zcela možné, neboť neexistují žádná univerzální data, jejichž výsledné hodnoty by tyto algoritmy univerzálně srovnávaly. Kromě toho hlavním cílem výpočtu v kontextu této práce není zjištění ideální hodnoty míry informace samotné, ale její porovnání s hodnotou vypočtenou na jiných datech.

Dále se vedou diskuse ohledně možné ztrátovosti komprese. Jak uvádí (Meinsma nedatováno), „neexistuje bezeztrátová komprese, která zmenší velikost bezvýhradně každého souboru“. (Cerra et al. 2010), zkoumající kompresi na základě analýzy fotografií Země z oběžné dráhy, používá bezeztrátovou kompresi. (Grünwald a Vitányi 2008) připouští rozšíření konceptu AIT i na ztrátové algoritmy, což „umožňuje formalizovat sofistikovanější pojmy, jako je „smysluplná informace“ a „užitečná informace“, které rozlišují, nakolik se v komprimovaných (nejčastěji mediálních) souborech nachází informace. Uvádí, že informace se může nacházet i v jejich šumech a jiných drobných nuancích, které lidský mozek nezaznamená vůbec, nebo pouze na nevědomé úrovni (dobrým příkladem je prostá komprese obrázků do formátu JPEG). Vzhledem k tomu, že neexistuje univerzální algoritmus ztrátové komprese pro veškeré typy souborů (obraz, zvuk, video apod.) a použití více různých algoritmů by výsledek neúměrně komplikovalo a zkreslovalo srovnání hodnot získaných různými algoritmy, pracuje nadále tato práce výhradně s bezeztrátovou kompresí.

Celkově tedy koncept AIT překonává omezení Shannonova přístupu, aniž by popíral jeho základní myšlenku. Zároveň je při použití dnešních technologií tato míra vypočitatelná velice snadno a tím dobře použitelná v této práci. Její hlavní nevýhodou je **výpočetní a kapacitní náročnost** při použití na velmi velkých datech a do jisté míry také **neexistence zcela univerzálního nejlepšího kompresního algoritmu**. V této práci bude pro výpočet používán kompresní algoritmus bzip2.

Použití pojmu „míra“ je na informační úrovni do jisté míry problematické, protože se názvem kryje se Shannonovým pojmem, který ovšem vyjadřuje spíše **opak redundance** datové reprezentace již dané informace.

Jedním z možných přístupů je měření délky nejkratšího textového vyjádření dotyčné informace. Zde se vyskytují dva problémy:

1. výsledná hodnota je zkreslena délkami názvů jednotlivých pojmů (a její hodnota by se tak například lišila pro různé jazyky, což je nežádoucí) a

2. neexistuje mnoho metod, jak toto textové vyjádření normalizovat, aby bylo maximálně stručné a nedošlo přitom ke ztrátě sdělení.

Možným řešením je normalizace textového vyjádření do formy trojic subjekt – predikát – objekt. To ovšem samo o sobě ještě neřeší první problém s délkami názvů, které beztak mají pouze identifikační funkci bez vlastního sémantického významu. Pro účely této práce je tedy **míra informace určena minimálním počtem trojic nutných k její reprezentaci**. Tento počet umožňuje hledat nejúspornější vyjádření a zároveň není vázaný na použití konkrétního jazyka nebo serializací, což by bylo problémem v případě užití délky textového vyjádření. Pakliže normalizace informace do množiny trojic není možná, lze také míru informace určit jako **nejmenší možnou velikost datové reprezentace**. Tento přístup opět úzce souvisí s algoritmickou teorií informace (AIT).

V literatuře nicméně lze narazit na jiný možný přístup, který používá přímo Shannonovu míru, kterou aplikuje na dané informační schéma (T-Box), nejčastěji vyjádřené jako ontologický slovník nebo ER diagram. Tento přístup je ovšem tedy v jistém rozporu se samotným pojmenováním „míra informace“, neboť se spíše blíží míře komplexity schématu informace, na druhou stranu i schéma jistou informaci nese a jedná se tedy o metriku, která by neměla být opomenuta.

Tento přístup reprezentuje (Doran et al. 2008), který v tradičním Shannonově vzorci nahrazuje pravděpodobnost p v zásadě nijak nesouvisejícím „stupněm vrcholu“ deg. Metrika neklade za cíl samotný výpočet míry informace a je entropií pouze inspirována; explicitním cílem je měření znuvupoužitelnosti a modularizovatelnosti ontologie. Stupeň vrcholu je měřen z grafu vizuální reprezentace ontologického slovníku, kde každý uzel v_i má jistý počet relací (v kontextu ontologií založených na jazyce OWL nejčastěji relace typu *subClassOf*), který je dělen součtem těchto hodnot pro veškeré ostatní uzly, tedy

$$p(v_i) = \frac{\deg(v_i)}{\sum_{v \in V} \deg(v)}.$$

Vzorec 3 - nahrazení pravděpodobnostní hodnoty ve výpočtu entropie podle (Doran et al. 2008)

Nevýhodou tohoto přístupu je jeho vazba pouze na úroveň T-Boxu (ačkoli použití na A-Boxu, v tomto případě na datech RDF nelze teoreticky vyloučit) a již zmíněné nahrazení

pravděpodobnosti. Na druhou stranu, ontologický slovník i obecně jakékoli schéma lze pokládat zároveň za množinu faktů o struktuře reality.

4.3 Jazyky matematické logiky

V případě jazyků matematické logiky, které umožňují **explicitně formalizovat informaci**, se do jisté míry jedná o strukturovanou metodu reprezentace informace, která ovšem není zpracovávána přímo strojově, avšak zaměřuje se na maximálně formalizované zprostředkování informace člověku. Konkrétně se jedná především o predikátovou logiku, vkládající n-tice proměnných (které zpravidla představují objekty světa) do relací, čímž je prakticky rozšířena výroková logika o vnitřní sémantický význam jednotlivých výroků.

Tyto relace mohou být unární ($R(x)$ značí vlastnost R objektu x), binární ($R(x, y)$ značí společnou vlastnost, tedy relaci nebo vztah objektů x a y) popřípadě pak jakéhokoli vyššího stupně. Za určitou formu podmnožiny lze považovat deskripční logiku, ačkoli dle (Svátek 2002) jde o „pojem zastřešující celou řadu příbuzných logických formalismů“, pro něž je typické zejména redukování stupňů relací pouze na dva a odstranění matematických funkčních operátorů.

Použití jazyků logiky může v obvyklých případech nahradit přístup konceptuálního datového modelování z předchozí části 4.3 s výhodou odstranění podvědomé případné vazby na datovou úroveň, stále ovšem setrvává omezení na strukturovanou informaci. Dále deskripční logika je technicky aplikovatelná za pomocí informačních ontologií, jimž se věnuje následující část.

Jazyky matematické logiky umožňují definovat tzv. **znalostní báze**. Za znalostní bázi označuje (Leslie F. Sikos, nedatováno) trojici skupin axiomů spadajících do tzv. „deskriptivní logiky prvního řádu“. Jedná se o

- **T-Box** – terminologické axiomy, skládající se z konceptů a jejich vztahů, z pohledu této práce dohromady tvořících jistou obdobu informačního schématu,
- **A-Box** – kolekce axiomů založených na individuích, z pohledu této práce obsahující konkrétní sdělení,
- **R-Box** – zařazení do rolí, nebývá vždy (zejména u ontologií) dle (Leslie F. Sikos nedatováno) uváděn.

V kontextu této práce je tedy pojem T-Box používán pro označení schématu informace. Zde je podstatné, že schéma nemusí vždy existovat, a to ani na informační, ani na datové

úrovni, kde je zpravidla závislé na použité datové technologii, proto typickým příkladem T-Boxu jsou ER modely, ontologické slovníky a formalizace pomocí výroků deskripční logiky. Jedním z problémů heterogenních dat (především z hlediska jejich strukturovanosti) přitom bývá absence samotného T-Boxu – neexistující schéma. A-Box je použit pro označení konkrétních faktů, týkajících se primárně popisované oblasti reality. Pojem R-Box zřejmě není pro tuto práci relevantní, a proto není dále uváděn.

4.4 Konceptuální datové modelování

Konceptuální datové modelování je procesem, při němž je **explicitně formalizována struktura** informace na úrovni T-Boxu (navzdory názvu, který ovšem vyjadřuje směřování celého procesu k datům, nikoli přímo data). Tento pojem bývá nejčastěji zmiňován pouze v souvislosti s procesem návrhu relační databáze prostřednictvím ER modelů a v praxi dokonce pouze jako součást vývoje konkrétní aplikace. To je nicméně velmi zúženým pohledem na problematiku tohoto procesu skládajícího se celkově ze čtyř úrovní, tvořících zároveň typické fáze průběhu datového modelování: konceptuální, logický, interní a externí, jak uvádí (Halpin 2001, s. 27). Přinejmenším modelování na konceptuální úrovni by tak správně mělo být považováno za modelování informační, nikoli datové.

Právě zmíněné zúžené chápání tohoto pojmu může v praxi způsobovat **nesystematický návrh** jiných než relačně-datových struktur (např. XML), kdy je fáze modelování informační úrovně a často i informace samotná upozaděna či zcela ignorována. To způsobuje problémy týkající se tří metrik touto prací navrhovaných. Zejména jde o možný vznik redundance v důsledku absence formalizovaných mechanismů normalizování dat nebo nevhodně nízkou úroveň strukturovanosti vytvářených dat.

ER (entity-relationship) model je jedním z nejtypičtějších nástrojů konceptuálního datového modelování. Jak je u nástrojů konceptuálního modelování běžné, je zaměřen čistě na modelování **strukturované** informace (o nejednoznačnosti tohoto pojmu pojednává sekce 4.2.1). Omezení tohoto typu jsou dle (Nečaský 2006) částečně řešitelná použitím rozšíření, jako je Extended E-R, EER, EreX a další.

4.5 Ontologie

Samotným principem informatického pojetí ontologií je explicitní (sdělitelná) formalizovaná **konceptualizace části světa** bez přímé orientace na datovou reprezentaci, čímž na

informační úrovni podobně jako jazyky logiky výrazně překonává relačně-databázově orientované konceptuální modelování, o němž mluví část 4.4. Například (Svátek 2002, s. 3) označuje informační ontologie za rozšíření konceptuálního modelování. Toto rozšíření slouží zejména k podpoře komunikace, a to na úrovni mezilidského porozumění, interoperability počítačových systémů a usnadnění aplikačního návrhu. Druhým důležitým možným využitím je publikování tzv. propojených dat na webu, umožňujících strojové zpracování.

5 Návrh datových metrik

V odborné literatuře existuje velký počet přístupů ke **klasifikaci metrik dat**, které se do různé míry překrývají. Zde je vhodné uvést dělení od například (Morton 2014, s. 4) postavené na konceptu Big Data, přinášející následující charakteristiky - objem (fyzické uložení), rozmanitost (zdroj a druh jednotlivých dat), rychlost (myšlena rychlost jejich produkce), hodnota (přínos pro organizaci) a pravdivost (nakolik data odpovídají reprezentované informaci).

Objem lze v kontextu metrik v této práci hodnotit spolu s ostatními metrikami coby prostou bitovou **velikost** zkoumaných dat (v případě textové podoby délku). Zohlednění objemu zároveň umožňuje metriky počítat váženým způsobem, kdy jejich hodnota je násobena touto velikostí – tento způsob výpočtu dále řeší kapitola 8.

Rozmanitost (heterogenita) dat je hlavní inspirací pro samotný vznik metrik navrhovaných touto prací. Práce ji nepoužívá coby samostatnou metriku, ovšem lze předpokládat, že rozmanitost dat může být ohodnocena na základě komplikovanosti metadatového skladu, o němž pojednává kapitola 8.

Rychlost produkce s metrikami v této práci příliš nesouvisí, ačkoli může být porovnána pomocí zaznamenávání časových údajů o době výpočtu metriky, jejíž hodnota se v čase může měnit. Dále je vhodné, když tato rychlost produkce je známa expertovi rozhodujícímu o případné periodicitě výpočet hodnot metrik.

Hodnota (přínos pro organizaci) není v kontextu metrik přímo zkoumatelná (jedná se zcela o měkkou metriku) a ani s nimi nesouvisí, neboť jsou zaměřeny spíše na vzájemné porovnávání dat a jejich struktur.

Pravdivost dat, navzdory její nesporné existenci může teoreticky být spolu s metrikami této práce měřena, takovéto měření ovšem nemá žádnou vazbu na v této práci akcentovanou problematiku kompatibility a je bližší samotnému obsahu dat a problému datové kvality.

Následující sekce podrobně rozebírají metodu výpočet tří metrik na datovém objektu (tento pojem je podrobněji definován v části 5.1). Výpočet je ve všech případech závislý na možnosti jednoznačně stanovit **velikost** datového objektu a jeho podobjektů. Její jednotka i přesný způsob stanovení se přitom odvíjí od typu zvolené datové struktury.

5.1 Objekt a elementární datový objekt

Datovým objektem je myšlena jednoznačně **ohraničená množina dat**, připravená na možný výpočet hodnot metrik. Pakliže dosud neexistuje nebo není podstatné její ohraničení a účel výpočtu, práce používá pouze pojem množina dat.

Objekt může být rozčlenitelný do menších (podřazených) datových objektů a být součástí nadřazených objektů. Objektem tedy je i celá zkoumaná datová základna, popřípadě i veškerá globálně existující data. Takovýto objekt na vyšší úrovni zpravidla reprezentuje konkrétní entitu, jakou je nejtypičtěji podnik, organizace nebo jejich stejně vymezené části. Tento typ hierarchie bývá také nazýván „kusovník“, přičemž anglický název „bill of materials“ (BOM) (BusinessDictionary nedatováno) indikuje použití především v průmyslové výrobě, ačkoli prakticky se jedná o typ rekurzivní datové struktury.

Elementární datový objekt je typem datového objektu – jde o nejmenší možný datový objekt, pro který stále je možný a smysluplný výpočet metrik a mající tak aktuální stav objektu definovatelný trojicí hodnot metrik. **Příkladem** je jedna buňka relační databáze, XML element obsahující pouze text nebo hodnota uvnitř JSONu neobsahující objekt ani pole.

Hlavní datový objekt (kořen stromové struktury) je naopak jediným a největším rozlišovaným datovým objektem – reprezentantem momentálně zkoumaných dat jako celku. **Příkladem** jsou zcela jakákoli data analyzovaná principy uvedenými v této práci.

Po praktické stránce tedy může být datovým objektem například

- databáze,
- její tabulka,
- řádek tabulky,
- XML dokument,
- XML element,
- soubor,
- adresář v souborovém systému,
- jakákoli jiná pojmenovatelná datová struktura (i ad-hoc účelově vytvořená, například za účelem transformace).

Datovým objektem je tedy i zcela účelově vytvořená kombinace dat, která má být jako celek zkoumána.

Datový objekt D může být charakterizován trojicí hodnot metrik: $s(D)$ – **míra strukturovanosti**, $h(D)$ – **míra hierarchičnosti** a $i(D)$ – **míra informace**. Doplňkovou hodnotou je jeho velikost $v(D)$, která není v práci chápána jako samostatná metrika, nýbrž součást výpočtu hodnot těchto tří metrik.

Analýzu datových úložišť principem jejich rozkladu řeší četné rámce zabývající se především **podnikovou architekturou**. Například TOGAF (The Open Group nedatováno) v sekci 9.3.1.2 „Identifikace požadovaných katalogů stavebních bloků dat“, kde představuje „hierarchické katalogy napříč vztaženými modelovými entitami“. Konkrétně jde o úrovně „logických datových komponent, fyzických datových komponent a datových entit“, přičemž vazba mezi logickou a fyzickou úrovní je blízká vztahu informační a datové úrovně představené touto prací. Toto pojetí je podstatně podrobnější, než jaké představuje tato práce a je navíc silně zaměřená na data v podniku. Členění na prostou hierarchii objektů v této práci je tedy zjednodušením, postačujícím pro účel výpočtu metrik a zkoumání kompatibility a přitom obecnější, použitelné i mimo oblast dat v podniku.

5.2 Míra strukturovanosti dat

Požadavkem na míru strukturovanosti je schopnost reagovat na 2 aspekty zkoumaných dat – jejich **schopnost rozkladu** na menší části a **míru strukturovanosti** jednotlivých **maximálně rozložených** částí dat (elementárních datových objektů). U elementárních datových podobjektů lze očekávat výrazně jednodušší způsob stanovení hodnoty této metriky. Oba tyto aspekty by měly výslednou hodnotu míry strukturovanosti zkoumaných dat zvyšovat.

Na rozdíl od informační úrovně je míra strukturovanosti na úrovni dat poměrně ustálené a zkoumané téma, byť jen málokdy přímo. Zejména je řešeno spíše z hlediska problému semistrukturovaných dat či nekompatibility různě strukturovaných dat. Také se často jedná spíše o nekompatibilitu již informace jimi reprezentované. Samotný pojem „strukturovanost“ je tedy používán minimálně.

Problém nestrukturovaných dat bývá často zmiňován v souvislosti se současnou narůstající rolí sociálních sítí, které jsou dle (Šperková 2014) jedním z jejich hlavních zdrojů.

Zde např. (Florescu 2005) představuje obecnou úvahu nad problémem méně strukturovaných informací (a tedy i dat) mimo jiné coby jednoho z **hlavních omezení RDBS** ve

svém klasickém pojetí. V určitém rozporu s tím (Helland 2017) chápe především semistrukturovaná data spíše negativně jako „nepříliš levný formát“ v důsledku „obalení“ samotných dat dalšími prvky nereprezentujícími informaci. (Cardelli 2001) se zabývá tvorbou deskripčního jazyka upraveného pro kontext semistrukturovaných, a navíc hierarchických dat[3]. Strukturovanost tak zpravidla není viděna coby spojitá metrika, nýbrž coby tři obecně uznávané úrovně strukturovanosti, jaké popisuje také např. (Bartmann et al. 2011):

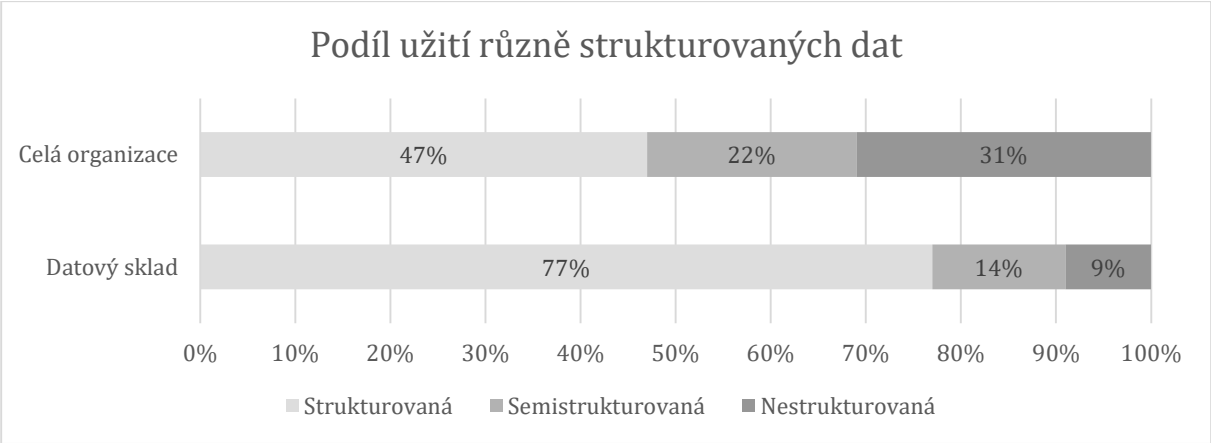
1. **Strukturovaná data** – data rozložitelná na atomické hodnoty a zcela normalizovaná či normalizovatelná, vytvářená, zpracovávaná a konzumovaná zejména strojově. Nejtypičtějším příkladem jsou normalizované RDBS v jejich základním pojetí dle (Codd 1990).

2. **Semistrukturovaná data** – data částečně obsahující strukturované i nestrukturované části, popřípadě pak části smíšené, kdy volný text získává částečnou strukturu jistým značkovacím mechanismem. Typickým příkladem je formát XML, ačkoli sám umožňuje libovolnou míru strukturovanosti, jak potvrzuje i (Tang et al. 2011), popřípadě příbuzný webový formát HTML (pro nějž jsou semistrukturovaná data zcela typická).

3. **Nestrukturovaná data** – zejména data vytvářená, zpracovávaná a konzumovaná člověkem, která nelze přímo (bez lidského zpracování nebo metod umělé inteligence) rozložit na atomické hodnoty při zachování sémantického významu. Typickým příkladem jsou prosté texty (abstrahujeme-li od případných kapitol a dalších semistrukturovaných prvků) a mediální soubory.

Podobného pojetí se drží i (Pokorný 2010) v souvislosti s tehdy aktuálním trendem, který posunul původní univerzální (myšleno klasické relační) pojetí databáze do datových prostorů, v nichž cílem je „správa bohaté kolekce strukturovaných, semistrukturovaných a nestrukturovaných dat“. Tehdejší podíl dat na těchto třech úrovních znázorňuje Obrázek 5. Aktuální průzkum na toto téma bohužel nebyl nalezen, nicméně konkrétní hodnoty podílů různě strukturovaných dat nejsou zcela podstatné. Tyto tři úrovně strukturovanosti zmiňuje i (Lněnička 2015, s. 111) coby důležitou charakteristiku při sběru dat v rámci užití metod Big data.

[3] kde mimo jiné do značné míry chybně mluví o datech, ačkoli deskripční jazyky a tím i jím zkoumaná oblast je na informační úrovni



Obrázek 5 – průzkum na 370 respondentech (Roberts 2010)

Chápání míry strukturovanosti v kontextu této práce je následující: míra strukturovanosti datového objektu je výsledkem rozložitelnosti dat na více menších, až elementárních datových podobjektů.

Dále je nutné brát v úvahu strukturovanost na úrovni konkrétními daty reprezentované informace (tedy v námi dostupné podobě v konkrétní situaci, jak popisuje sekce 4.2.1). Zde lze odvodit, že informace mající určitou míru strukturovanosti může být přímo reprezentována pouze daty stejně nebo méně strukturovanými.

Tento koncept je pravděpodobně nejsnáze aplikovatelný na formát XML, který přímo jednotlivé úrovně strukturovanosti podporuje. Nelze vyloučit ani aplikace na RDBS, které umožňují uložení nestrukturovaných dat, ačkoli v rozporu se základními principy normalizace, či i uložení přímo XML, jak ukazují technologie XML databází a jak studuje i např. (Krishnamurthy et al. 2001). Tímto výpočtem míry strukturovanosti se zabývá například (Tang et al. 2011), užívající k výpočtu na formátu XML již zmíněnou entropii, která nahrazuje hodnotu pravděpodobnosti výpočtem $p(X = x_i) = \frac{n(X=x_i)}{N}$, kde N je počet veškerých cest uvnitř XML dokumentu a $n(X = x_i)$ je počet cest shodných jako úsek x_i. Tento přístup nicméně je v rozporu s pojetím míry strukturovanosti v této části a má blíže k samotné míře informace či výpočtu redundance.

Ačkoli bylo zmíněno, že míra strukturovanosti bývá rozlišována na třech úrovních, je přesto možné ji uvažovat coby **spojitou metriku**, jejíž hodnota $s(D) \in \langle 0; 1 \rangle$ kde s je **funkce** samotné míry strukturovanosti, D značí **datový objekt**. Předpokládáme, že tento datový objekt je rozložitelný na **nejmenší části** (elementární podobjekty) $c_1, \dots, c_n (n \geq 1)$, přičemž možný rozklad na větší objekty na vyšší než elementární úrovni hierarchie není pro výsledek podstatný. Na tomto datovém objektu D bude míra strukturovanosti s

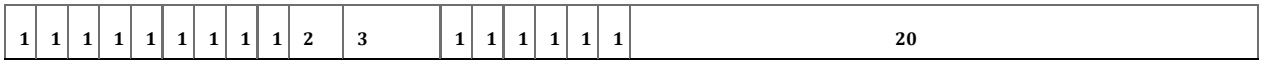
vypočtena na základě hodnot odpovídajících jeho elementárním podobjektům o své určité velikosti: hodnota $s(c_j) = 0$ odpovídá zcela **nestrukturovanému** elementárnímu datovému podobjektu, zatímco $s(c_j) = 1$ odpovídá zcela **strukturovanému** elementárnímu datovému podobjektu. Dále může u elementárních podobjektů teoreticky nastat i situace, kdy míru strukturovanosti nelze jednoznačně určit. Zde je možné použít hodnotu $s(c_j) = 0{,}5$ a předpokládat, že se jedná o semistrukturovaný datový objekt, nicméně tento výjimečný případ nebude dále v práci uvažován.

Následující příklady znázorňují několik situací, které by vzorec pro výpočet míry strukturovanosti měl dokázat zohlednit tak, aby jeho hodnota byla blízká požadovanému výsledku.

Pro názornost následujících příkladů je použita grafická reprezentace, kde datový objekt je reprezentován řadou atomických podobjektů (představovaných jednotlivými obdélníky), na které může být rozložen. **Bílé obdélníky** značí podobjekty nestrukturované (hodnota míry strukturovanosti 0), **černé obdélníky** pak objekty strukturované (hodnota míry strukturovanosti 1). Jejich relativní velikost v rámci celého datového objektu vyjadřuje uvedené číslo a šířka.

Příklad 5: Uvažujme relační databázi ukládající data z účetnictví. Prakticky výhradně jde o číselné údaje, které jsou pouze výjimečně doplněné popisem skládajícím se z věty. Tyto data lze bezpochyby označit za velmi strukturovaná – **je očekávána hodnota míry strukturovanosti blízká jedné**.

Příklad 6: Uvažujme web, obsahující přímé texty řady dlouhých knih, u nichž nelze najít vnitřní strukturu a nemají formálně oddělené ani kapitoly (jsou například pouze oskenovány). Jednotlivé knihy se ovšem nachází na oddělených stránkách webu. Pro jednoduchost je předpokládáno, že jsou všechny stejně velké, pouze jedna je větší dvakrát, jedna třikrát než ostatní a jedna je výrazně, dvacetinásobně větší. Ačkoli jsou tak zcela nestrukturované, jistá struktura vzniká jejich oddělením a seřazením. Tato data jsou strukturovaná částečně – **je očekávána hodnota míry strukturovanosti blízká 0,75**, rostoucí s počtem knih, tedy s počtem elementárních datových podobjektů.



Příklad 7: Uvažujme jeden velký nestrukturovaný soubor, například video a vedle něj několik drobných položek metadat – u nich lze očekávat, že míru strukturovanosti zvýší, ovšem vzhledem ke svému minimálnímu podílu velikosti zanedbatelným způsobem - **je očekávána hodnota míry strukturovanosti blízká nule**.

50	1	1

Vzorec pro výpočet míry strukturovanosti tedy vychází z následujících předpokladů:

- u elementárního podobjektu lze určit míru strukturovanosti ($s(c_j)$) jednoznačným triviálním, konkrétním a automatizovatelným způsobem (např. zda u textu jde o větu, zda jde o přirozeně nestrukturovaný, například mediální formát apod.) a její hodnota je binární, tedy 1 nebo 0,
- na výslednou míru strukturovanosti má vliv míra strukturovanosti a počet jednotlivých elementárních podobjektů.

Je tedy zřejmé, že **není možné aplikovat** při výpočtu prostý **vážený průměr** míry strukturovanosti elementárních podobjektů – pro **Příklad 6** by výslednou hodnotou míry strukturovanosti bylo $s(D) = 0$ a možnost dělení objektu na menší podobjekty by byla nelogicky ignorována.

Velikost dat je značena v a určena rozměrem v bitech, popřípadě u čistě textových formátů pro zjednodušení jejich prostou délkou. Podíl velikosti elementárního datového podobjektu c_j na celém datovém objektu D, tj. $\frac{v(c_j)}{v(D)}$ je nazván **váha elementárního podobjektu** a jde o nejvyšší hodnotu, o níž teoreticky může podobjekt snížit celkovou hodnotu strukturovanosti.

Celková hodnota $s(D)$ vychází z vah jednotlivých elementárních podobjektů. Hodnota sčítance v rámci celkové sumy vychází z váhy elementárního podobjektu, která je u nestrukturovaných elementárních podobjektů ($s(c_j) = 0$) snížena, ovšem ne až na nulu (což by fakticky odpovídalo výpočtu váženého průměru, viz dále konstanta k), jen je násobena koeficientem s hodnotou $1 - \frac{v(c_j)}{v(D)}$, který vyjadřuje podíl na rozložitelnosti celého datového objektu. U strukturovaných elementárních podobjektů ($s(c_j) = 1$) je hodnotou tohoto sčítance celá váha dotyčného podobjektu.

Celkový navržený výpočet míry strukturovanosti s datového objektu D v této práci definuje Vzorec 4.

$$s(D) = \sum_{j=1}^{n} \left(1 - \frac{v(c_j)}{v(D)}\left(1 - s(c_j)\right)\right) \cdot \frac{v(c_j)}{v(D)}$$

Vzorec 4 – výpočet míry strukturovanosti

Takovéto nastavení vzorce znamená, že například datový objekt skládající se ze dvou **stejně velkých** podobjektů (jednoho strukturovaného a jednoho nestrukturovaného, oba mají tedy váhu $\frac{1}{2}$) má míru strukturovanosti nikoli 0,5, jak by odpovídalo použití váženého průměru, ale

$$s(D) = \left(1 - \frac{1}{2}(1 - 1)\right) \cdot \frac{1}{2} + \left(1 - \frac{1}{2}(1 - 0)\right) \cdot \frac{1}{2} = 0{,}5 + 0{,}25 = 0{,}75,$$

neboť se projevil vliv, že objekt je dělitelný na dva elementární podobjekty a nestrukturovaný elementární datový podobjekt tak snížil hodnotu pouze o 0,25.

Zde je možné poznamenat, že poměr **vlivu dělitelnosti** objektu a **vlivu** samotné **strukturovanosti** elementárních podobjektů by bylo vhodné moci nastavit. Vzorec lze za tímto účelem zobecnit tak, že váha elementárního podobjektu snižující celkovou míru strukturovanosti je umocněna na konstantu k, čili

$$s_k(D) = \sum_{j=1}^{n} \left(1 - \left(\frac{v(c_j)}{v(D)}\right)^k \left(1 - s(c_j)\right)\right) \cdot \frac{v(c_j)}{v(D)},$$

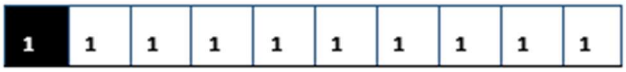
Vzorec 5 – výpočet míry strukturovanosti zobecněný o konstantu vlivu dělitelnosti objektu

kde platí, že $k > 0$, přičemž hodnota $k = 1$ je fakticky původní nastavení vzorce, zatímco hodnota $k = 0$ činí z vzorce již zmíněný vážený průměr. Konstanta $\boldsymbol{k < 1}$ **snižuje vliv dělitelnosti** na hodnotu strukturovanosti, zatímco $\boldsymbol{k > 1}$ **ji zvyšuje**.

Nastavení konstanty k je porovnáno na šesti případech, z toho jde o **tři varianty rozložení elementárních podobjektů** stejné velikosti a dále výše v této části znázorněných tři **příklady** (Příklad 5 - Příklad 7). Tři doplněné varianty ukazují různé rozložení strukturovaných a nestrukturovaných elementárních datových podobjektů. Jejich podoba (při použití stejného grafického znázornění jako v příkladech výše v této kapitole 5.2) je následující:

1.  - většina strukturovaných,

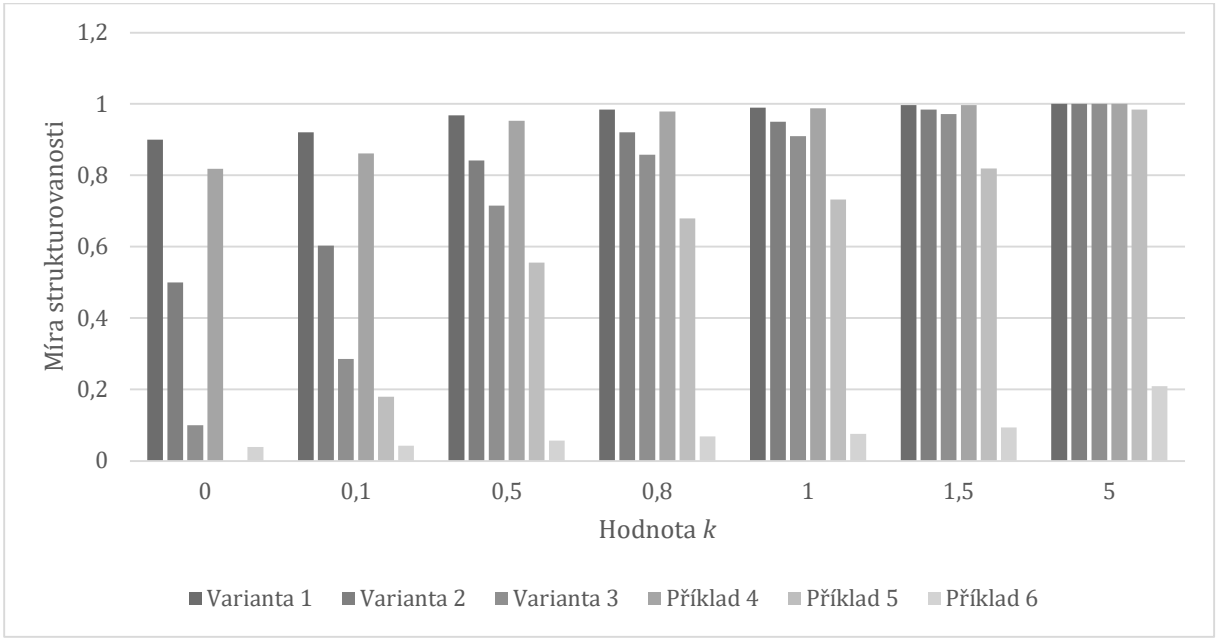
2.  - rovnoměrné rozložení,

3.  - většina nestrukturovaných,

přičemž je porovnáno 6 různých hodnot konstanty k. Výsledek ukazuje Tabulka 1.

k	0	0,1	0,5	0,8	1	1,5	5
Varianta 1	0,9000000	0,9205672	0,9683772	0,9841511	0,9900000	0,9968377	0,9999990
Varianta 2	0,5000000	0,6028359	0,8418861	0,9207553	0,9500000	0,9841886	0,9999950
Varianta 3	0,1000000	0,2851046	0,7153950	0,8573596	0,9100000	0,9715395	0,9999910
Příklad 5	0,8181818	0,8617366	0,9532080	0,9789819	0,9876033	0,9966272	0,9999994
Příklad 6	0,0000000	0,1792280	0,5554340	0,6792253	0,7325000	0,8196415	0,9843748
Příklad 7	0,0384615	0,0422254	0,0571340	0,0681629	0,0754438	0,0933980	0,2096855

Tabulka 1 – srovnání vlivu konstanty k na výsledek hodnoty míry strukturovanosti s_k



Obrázek 6 – graf vlivu konstanty k na výsledek hodnoty míry strukturovanosti s_k

Sloupec v tabulce pro hodnotu $k = 0$ ukazuje fakticky vážený podíl strukturovaných podobjektů s nulovým vlivem dělitelnosti. Dále sloupec $k = 1$ obsahuje hodnoty původní podoby vzorce (viz Vzorec 4), kdy je vliv hodnot míry strukturovanosti podobjektů a dělitelnosti rovnoměrný. Pro nižší hodnoty k je vidět, že dělitelnost má nižší vliv na výsledek míry strukturovanosti. Naopak pro vyšší hodnoty k platí, že s jejich růstem výsledek $s_k(D)$ jde k jedné ($\lim_{k\to\infty} s_k(D) = 1$). Zřejmě tedy **neexistuje důvod hodnoty $k > 1$ používat**.

Obrázek 6 vizuálně potvrzuje, že **nižší hodnoty** k dokážou **lépe rozlišit** výsledné hodnoty míry strukturovanosti. Z několika ukázkových příkladů ovšem nelze jednoznačně vyvodit, že vzájemně odlišnější hodnoty $s_k(D)$ jsou vhodnější. Zde se do budoucna nabízí možnost porovnat hodnoty s jinou mírou strukturovanosti, získanou například na základě dotazníku a stanovení optimální hodnoty k. Proto tato práce v analýze dat v kapitole 10 používá hodnotu $\boldsymbol{k = 1}$.

Dále lze k navržené funkci míry strukturovanosti (Vzorec 4 a Vzorec 5) také podotknout, že způsob výpočtu na základě bitové velikosti hrozí neúměrným „zveličením“ nestrukturovaných dat zejména u mediálních souborů, kde lze očekávat výrazně vyšší velikost než u textů. Na druhou stranu právě samotná velikost může být při analýze dat velmi zásadní. Tento rozpor je vhodné dále v práci vyřešit praktickým testováním.

Také je podstatné, že rozdělení jakýchkoli dat na elementární podobjekty nemusí vždy znamenat, že data jsou rozložena veškerá. Například u textově orientovaných formátů (XML, JSON apod.) se mohou vyskytovat bílé znaky, nereprezentující žádnou informaci, avšak jsoucí součástí těchto dat. Ty mohou ovlivňovat výpočet kvůli zvětšování velikosti celých dat ($v(D)$) a tím snižovat vliv méně strukturovaných podobjektů na pokles celkové míry strukturovanosti dat ($\frac{v(c_n)}{v(D)}$) a tím vytvářet hodnotu míry strukturovanosti neúměrně vysokou. V těchto případech je vhodné nahradit velikost celého objektu $v(D)$ sumou velikosti dotyčných podobjektů $\sum_{j=1}^{n} v(c_j)$. Takto upravený Vzorec 4 je aplikován v sekci 10.2.3.

5.3 Míra hierarchičnosti dat

Požadavkem na metriku míry hierarchičnosti je vyjádření **podílu velikosti** takových částí zkoumaných dat, která vyvolávají **porušení definice hierarchičnosti** a v případě jejich hierarchizace lze očekávat, že bude nutné je duplikovat.

Definice hierarchičnosti se v případě datové úrovně shoduje s úrovní informační a definicí hierarchičnosti na straně 29. To ovšem neznamená, že hierarchičnost informace musí být shodná s mírou hierarchičnosti ji reprezentujících dat.

Nesoulad hierarchičnosti mezi informací a daty má tři hlavní příčiny. **První příčinou** je přirozená tendence člověka vnímat modelovanou realitu hierarchicky, pokud se jí zabývá pouze z jediného hlediska, jak popisuje podrobněji sekce 4.2.2, což se projevuje i na člověkem vytvořených datových modelech. **Druhou příčinou** je fakt, že na datové úrovni je možné počítat hierarchičnost nejen pouze ze schématu, nýbrž přímo z dat na úrovni A-Boxu. Jejich velikost napříč schématem může být libovolně různorodá a velikost podílu různě hierarchických dat má vliv na celkovou hodnotu míry hierarchičnosti, jak dále ukazuje Vzorec 6. **Třetí příčinou** je častá nutnost pracovat s typem datové struktury, která jiné než hierarchické uložení nepodporuje, nebo ho podporuje nevhodným či nejednoznačným způsobem (formáty XML, JSON apod.). Problém hierarchizace dat dále řeší sekce 7.2.3.

Problém cyklu se shoduje se stejným problémem jako na úrovni informace – schéma ho obsahovat může (příkladem je opět kusovník), ovšem orientovaný graf reprezentující množinu dat coby vzájemně odkazujících datových objektů nikoli, neboť takto zacyklená data hierarchicky reprezentovat nelze.

Na úrovni dat je tedy míra hierarchičnosti úměrná převoditelnosti dat do hierarchické podoby s minimální nutností části převedených dat opakovat a vytvářet tak redundanci. Podobně jako v případě míry strukturovanosti vychází i míra hierarchičnosti z rozdělení analyzovaného datového objektu D na n elementárních datových podobjektů $c_1, \dots, c_n$. Připomeňme, že na podobjektech datového objektu D je definován predikát hierarchické vlastnosti H (viz sekce 4.2.2) na základě relace R, která vyjadřuje vzájemnou podřízenost podobjektů. Z elementárních datových podobjektů lze vybrat pouze ty podobjekty $c_1^H, \dots, c_m^H$, které splňují predikát H a **nejsou podřízené žádnému objektu nesplňujícímu predikát hierarchičnosti** H uvnitř celého zkoumaného objektu D (a v případě hierarchizace dotyčných dat nedojde k jejich znásobení a tím vzniku redundance).

$$h(D) = \frac{\sum_{i=1}^{m} v(c_i^H)}{v(D)}$$

Vzorec 6 – výpočet míry hierarchičnosti

Vzorec 6 definuje celkový výpočet míry hierarchičnosti h datového objektu D coby podíl **sumy velikosti** v hierarchických (definici splňujících) elementárních datových podobjektů c_i^H a celkové velikosti datového objektu $v(D)$. Hodnota $v(D)$ může být stejně jako v případě míry strukturovanosti po zvážení nahrazena sumou velikostí všech elementárních podobjektů, označenou stejně jako v případě výpočtu míry strukturovanosti hodnotou $\sum_{j=1}^{n} v(c_j)$, a to ze stejných důvodů jako v tomto předchozím případě (sekce 5.2).

5.4 Míra informace a redundance dat

Požadavkem na metriku míry informace je pokles její hodnoty v případě, kdy zkoumaná data obsahují opakovaná sdělení.

Jak poznamenala sekce 4.2.3, míra informace je nižší ve chvíli, kdy stoupá pravděpodobnost přijetí shodného sdělení (reprezentujícího stejnou informaci), které je tedy redundantní. Míru informace v datech lze poté považovat za míru opačnou (inverzní) oproti míře redundance.

Míra informace v tomto chápání (opak redundance) je tedy metrikou týkající se **pouze datové úrovně**, neboť fakticky vyjadřuje míru „neopakovatelnosti údajů" datové reprezentace informace. Datová množina dle (Clark et al. 2015) je konzistentní, „když každá dvojice případů se stejnými hodnotami atributů patří do stejného konceptu", což fakticky vylučuje, aby informace zpětně odvozená z dotyčných dat obsahovala kontradikci. Datová konzistence je pojmem známým zejména z databázové teorie, kde se stává synonymem plnění integritních omezení, která jsou součástí kvalitně navržené databáze, která možná porušení konzistence minimalizuje.

Redundance je nejtypičtějším a nejzásadnějším zdrojem možného porušení konzistence, kdy je konkrétní fakt (informace) ukládaný v datech vícenásobně (Halpin 2001, s. 112), což přináší specifické problémy – znásobení nákladů na správu takovýchto dat, spojených s větší velikostí, náročnějším zabezpečením a náročnějším zajištěním konzistence. Zde ovšem jde o redundanci na úrovni schématu dat, tedy faktické umožnění existence redundantních dat jejich schématem, které nemusí implikovat reálnou existenci redundantních dat.

I z hlediska měření míry informace lze analyzovat **data samotná** (což je vzhledem k zařazení této kapitoly hlavním cílem), popřípadě jejich **schéma** nebo kombinaci obojího. U

schématu však nedává pojem redundance v tomto kontextu smysl (její význam je v případě datového modelování do jisté míry odlišný), ovšem výpočty založené na vzorci entropie možné jsou). Přístup zkoumající přímo dat užívá (Ruellan 2012), ovšem čistě na základě technické zpracovatelnosti dokumentu, což o způsobu reprezentace informace příliš nevypovídá. Schémata naopak zohledňuje (Thaw a Khin 2013) coby míru kvality jejich návrhu dle „míry komplexity schématu“ zkoumané v (Thaw a Khin 2011). Podobně (Basci a Misra 2011) užívá komplexitu schématu, ovšem v souvislosti s „udržovatelností systému“. Podobně dále (Feuerlicht et al. 2015) měří entropii, ovšem za účelem zkoumání komplexity doménových standardů, specifikovaných právě pomocí schémat XML. S tím souvisí i podmíněná entropie schémat (tentokrát zaměřených obecně, i mimo kontext XML) navržená v (Arenas a Libkin 2005).

Výpočet této hodnoty přímo na schématu rozlišuje dva přístupy: **měření kvalitativních charakteristik** schématu samotného, jak je uvedeno výše v předchozím odstavci, a dále **predikci pravděpodobné kvality** předpokládaných schématu odpovídajících dat; podobný průzkum ovšem nebyl doposud s největší pravděpodobností realizován.

Podobně jako hierarchičnost je i redundance zkoumatelná zejména na strukturovaných datech, což částečně řeší problematika již uvedené AIT při zvoleném kompresním algoritmu. Při její aplikaci je ovšem nutné předpokládat, že opakující se fragmenty dat jsou vždy redundancí, což nemusí být pravdivé. Redundance přitom nemusí být v každém případě negativním jevem; existuje řada případů, kdy je cíleně vytvářena. Nejtypičtějším takovýmto případem je **rozdělení dat na transakční a analytická** (OLTP a OLAP) a jejich ukládání do mezipaměti (cache) za účelem zrychlení přístupu. Zde často jde o data, která nejsou přímo duplicitně ukládána, ovšem jsou odvozena z jiných uložených dat, a to často na základě komplikovaných výpočtů a transformací. V takových případech je do jisté míry sporné, zda a nakolik se jedná o redundanci.

Dále lze redundanci rozlišit na vnitřní, **interní** (vícenásobné úložiště v rámci jedné množiny dat) a vnější, **externí** (úložiště je duplikováno mimo zkoumanou množinu dat, často například v rámci cloudové aplikace). Externí redundance může být zdůvodněna obdobným způsobem jako případ v předchozím odstavci a nemusí být vypočitatelná ani jinak zjistitelná, proto nebude dále uvažována.

U datového elementárního objektu lze tímto způsobem rozlišovat interní a externí redundanci obdobně, ovšem u velmi malých elementárních datových objektů je interní redundance prakticky vyloučena.

Vnější redundanci r_e datového elementárního objektu D lze spočítat jako prostý počet dalších existujících úložišť reprezentujících stejnou informaci (nikoli nutně obsahujících stejnou hodnotu); jedná se tedy o celé nezáporné číslo. **Vnitřní redundance** může být spočítána coby prostý kompresní poměr na základě AIT a je aplikována i v praktických výpočtech v kapitole 10.

$$r(D) = \frac{1 + r_e(D)}{\frac{v(D)}{v\big(c(D)\big)}} = \frac{\big(1 + r_e(D)\big)v\big(c(D)\big)}{v(D)}$$

Vzorec 7 – výpočet vnitřní a vnější redundance datového elementárního objektu

Vzorec 7 uvádí výpočet redundance. Funkce r je výsledná redundance, D je datový objekt, r_e je vnější (externí) redundance objektu (počet dalších úložiště stejné informace), v je funkce bitové velikosti a c je funkce použité bezeztrátové komprese. Vnější redundance (která je nulová při neexistenci dalšího úložiště) je zvýšena o 1, aby bylo možné ji kombinovat s kompresním poměrem (na intervalu od nuly do jedné), který byl použit, aby hodnota nebyla závislá na velikosti datového elementárního objektu.

Dále je nutné podotknout, že vnější redundance a tím i další úložiště mohou být z vzorce vypuštěny z výše uvedených důvodů případného obtížného až nemožného zjištění a případná transformace dat v rámci datového elementárního objektu na vnější redundanci nemá vliv, tedy nemusí být ani objektem zájmu. V tomto případě lze míru informace redukovat na prostý podíl velikosti komprimovaných a nekomprimovaných dat, a sice

$$r(D) = 1 - \frac{v\big(c(D)\big)}{c(D)}$$

Vzorec 8 – výpočet interní redundance datového objektu

Tímto svým pojetím, způsobem výpočtu a vazbou na míru informace lze redundanci považovat za jistou datovou analogii míry informace na informační úrovni, kterou zkoumá sekce 4.2.3. Míra informace i datového objektu D je, jak již bylo uvedeno, v tomto pojetí opačnou mírou vůči redundanci, a definuje ji Vzorec 9.

$$i(D) = 1 - r(D) = \frac{v\big(c(D)\big)}{v(D)}.$$

Vzorec 9 – výsledný výpočet algoritmické míry informace datového objektu

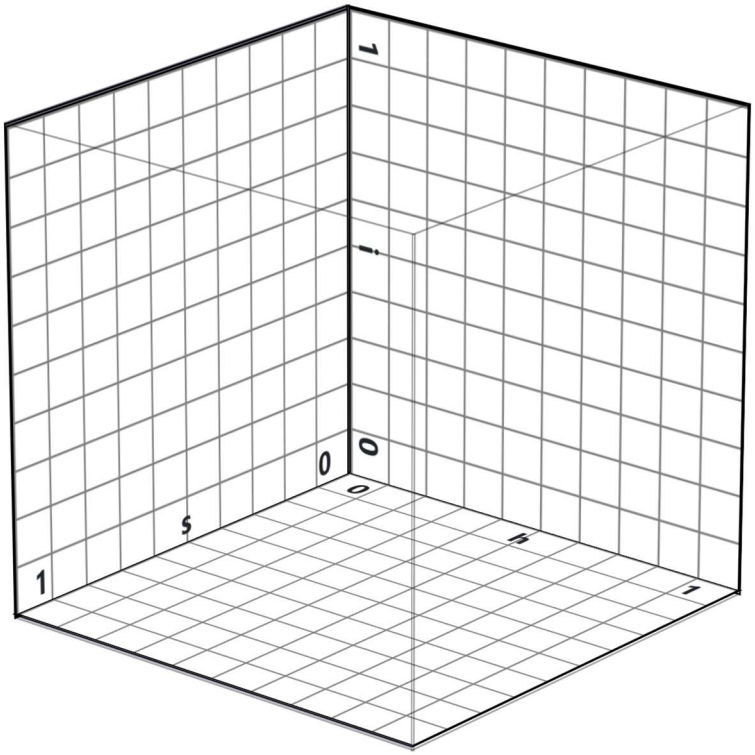
Dále v této práci je tedy mírou informace myšlen tento **nejnižší dosažitelný kompresní poměr při daném použitém kompresním algoritmu**, který se blíží spíše „hustotě“ informace v datech. Samotnou mírou informace by tak teoreticky také mohla být pouze prostá velikost komprimovaných dat. Vzhledem k tomu, že velikost komprimovaných dat a tedy ani výsledná hodnota pochopitelně nemůže být nulová, je míra informace v této práci oproti oběma dalším metrikám škálována na zleva otevřený interval $(0; 1\rangle$ – není možné zkomprimovat data o nenulové velikosti $v(D)$ na nulovou velikost $v\big(c(D)\big)$.

5.5 Možnosti vizualizace měřených hodnot

5.5.1 Trojrozměrný metadatový prostor

Vizualizační metody užívané v této práci chápou hodnoty **metrik datového objektu coby souřadnice geometrického bodu** v již uvedeném trojrozměrném metadatovém prostoru, kde 3 osy (lze teoreticky použít i jiný počet) reprezentují vybrané metriky. Objektů může být v jednom čase zobrazeno v souladu s jejich definicí v části 5.1 neomezený počet. Při vysokém počtu objektů nicméně nemusí být body v prostoru vizuálně přehledné a je vhodné nahradit zobrazení samotných bodů zobrazením **jimi pokrytých oblastí** o tvaru krychle, jejíž hrana je zlomkem hodnoty 1, tedy hrany celého krychlového trojrozměrného prostoru.

Dále je možné znázorňovat změny v trojrozměrném metadatovém prostoru, tedy datové transformace, které detailněji popisuje část 7.3 a to prostřednictvím mnoha přístupů, například prosté šipky, jak dále ukazuje Obrázek 19.



Obrázek 7 – trojrozměrný metadatový prostor pro účel vizualizace hodnot metrik

Samotný (pro názornost prázdný) prostor ukazuje Obrázek 7. Trojrozměrná vizualizační metoda umožňuje znázorňovat širokou škálu teoretických aspektů této práce. Její další výhodou je možnost snadno zkoumat vzájemnou informačně-datovou kompatibilitu více datových objektů či typů datových struktur na základě průniku jejich geometrických reprezentací.

Trojrozměrný vizualizační přístup je bezpochyby použitelný i v rámci metadatového skladu. Aktuálně vyfiltrované hodnoty mohou být místo klasického grafu okamžitě zobrazovány v trojrozměrném metadatovém prostoru. Hlavní výhodou oproti běžným dvojrozměrným grafům a kontingenční tabulce je právě možnost zkoumání hodnot tří metrik najednou.

Trojrozměrný metadatový prostor ovšem může být limitující pro následné rozšiřování počtu metrik a prakticky i limituje znázornění prosté velikosti dat, tedy jejich pomyslné čtvrté metriky. Potenciální další rozměr lze doplnit například již zmíněnou animací, tedy časem, barvou jednotlivých entit. Také lze použít i velikost těchto entit (krychlí), což dává smysl především při znázorňování datové velikosti, ovšem zároveň toto z geometrického hlediska staví jistou názornost na úkor přesnosti zobrazení.

5.5.2 Vizualizace typů datových struktur a datových objektů

Typ datové struktury (některý z formátů a typů datových modelů uvedených v kapitole 6 nebo i případně jiný) je možné vizualizovat coby geometrický útvar **pokrývající části trojrozměrného metadatového prostoru**, v nichž se mohou **v optimálním případě** vyskytovat struktuře odpovídající datové objekty. Princip vizualizace typu datové struktury je prakticky aplikován v kapitole 6.

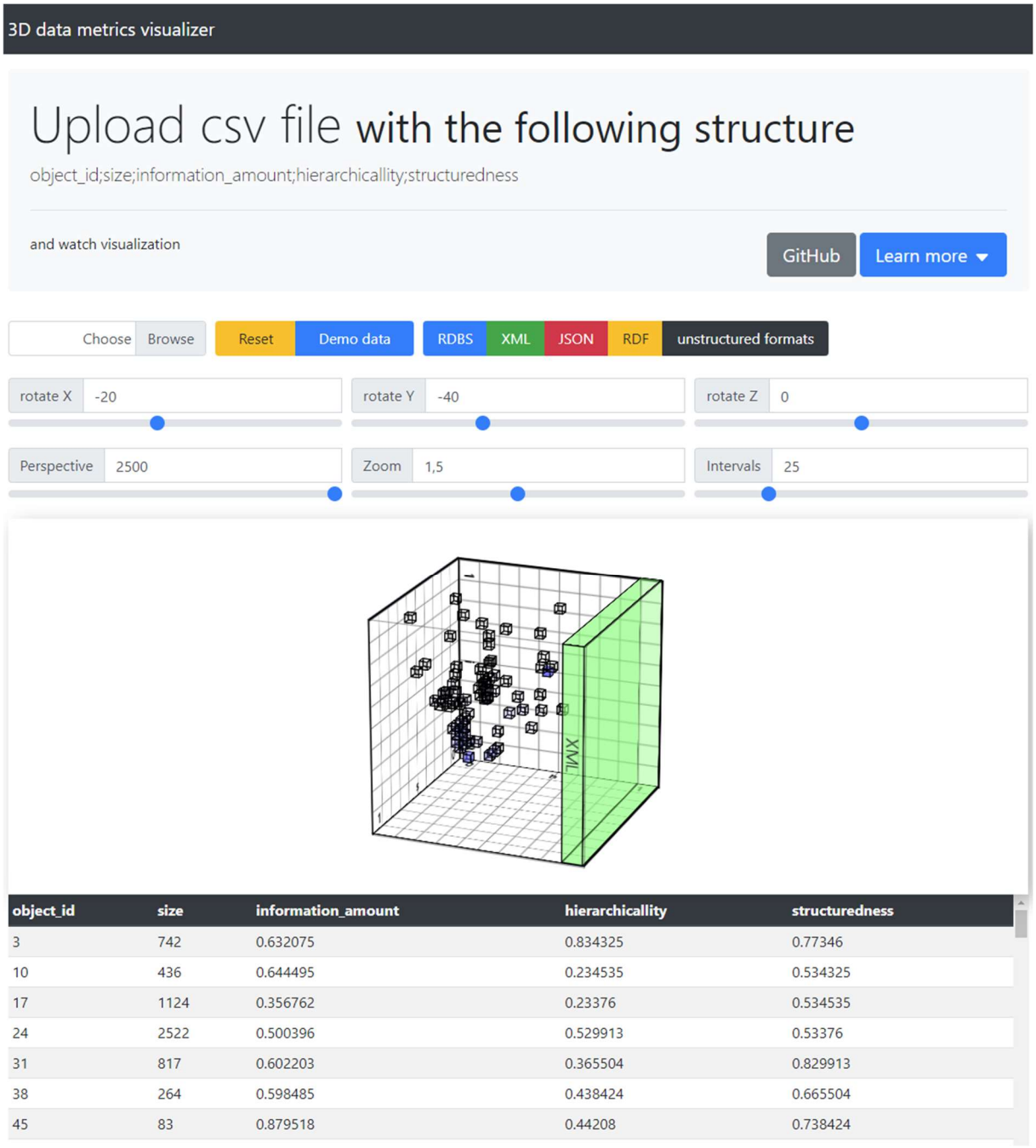
Pakliže jsou rozsahy **ideálních** hodnot metrik pro daný typ datové struktury určeny trojicí prostých intervalů, je takový typ datové struktury vizualizován kvádrem. Nelze ovšem vyloučit, že v případě dalších, prací nezkoumaných typů datových struktur se může jednat i o jiné, komplexnější geometrické útvary.

Každý datový objekt je charakterizován hodnotami s – míra strukturovanosti, h - míra hierarchičnosti, i – míra informace a geometricky je tak reprezentován **bodem majícím dotyčné souřadnice**. Protože vizualizovat v prostoru vyšší počet bodů je nepříliš názorné, lze body nahradit krychlemi na místech, do nichž některé z bodů zasahují, jak je prakticky ukázáno v kapitole 10.

5.5.3 Nástroj 3D data metrics visualizer

Tato autorem webová vyvinutá aplikace (Daniel Vodňanský 2020) slouží k tvorbě trojrozměrných vizualizací použitých v této práci. Jejím vstupem je CSV soubor obsahující seznam datových objektů a jejich charakteristik. Dále je možné vložit kvádry zachycující prostor pokrytý jednotlivými typy datových metrik.

Celý prostor lze otáčet po třech osách, měnit úhel perspektivy, zvětšovat, a především pak určit velikost zlomku intervalu pokrytých oblastí prostoru, tedy malých krychlí, v nichž jsou obsaženy souřadnice datových objektů. Samotná vizualizace je v souladu s principy popsanými předchozími sekcemi 5.5.1 a 5.5.2. Náhled aplikace s ilustračními datovými objekty ukazuje Obrázek 8. Samotná aplikace je vystavena na adrese https://danielvodnansky.github.io/3d-data-histogram/.



object_id	size	information_amount	hierarchicallity	structuredness
3	742	0.632075	0.834325	0.77346
10	436	0.644495	0.234535	0.534325
17	1124	0.356762	0.23376	0.534535
24	2522	0.500396	0.529913	0.53376
31	817	0.602203	0.365504	0.829913
38	264	0.598485	0.438424	0.665504
45	83	0.879518	0.44208	0.738424

Obrázek 8 – nástroj 3D data metrics visualizer (Daniel Vodňanský 2020)

6 Typy datových struktur

V současné době velikost i **heterogenita datových úložišť**, jejich formátů a modelů, a i dat v nich obsažených patrně i vzhledem k technologickému rozvoji narůstá. Praktickým důsledkem této výrazné datové heterogenity nejen podnikových dat je často nesystematický a nekoncepční přístup k návrhu architektury datové základny. Ten může spočívat především v užití řady nekompatibilních a neintegrovaných datových úložišť, přinášejících zejména značnou míru redundance. S ní se zbytečně zvyšuje objem dat a zpomaluje proces jejich zpracování, čímž je činí hůře použitelnými, a tedy i méně hodnotnými. Dále podobný přístup komplikuje a tím zhoršuje možnosti zabezpečení dat. O to více pak v tomto ohledu „trpí" především menší nežli velké firmy s nižší formalitou řízení a horší dostupností odborné IT expertízy. Hlavní příčinou je, že většina modelů řízení podnikové informatiky, jako jsou ITIL nebo COBIT sice dodávají i řízení dat určitý rámec, k němuž podobně jako tato práce poskytují řady metrik, jsou ovšem pro použití v sektoru malých a středních podniků příliš komplexní. Na druhou stranu je jistou výhodou sektoru SME možná nižší komplexita i velikost samotné datové základny, která případnou expertízu nepochybně usnadňuje.

V souvislosti s tímto fenoménem bývá často zmiňován koncept nazývaný **Big Data**. Zde je nutné poznamenat, že pravděpodobně neexistuje všeobecně přijímaná definice tohoto pojmu, jak potvrzuje i (Morton 2014). Například (Gartner 2017) uvádí, že „Big data jsou vysokoobjemová, vysokorychlostní a vysoce rozmanitá informační aktiva vyžadující nákladově efektivní inovativní formy zpracování informací, která umožňují lepší pochopení, tvorbu rozhodnutí a procesní automatizaci". Fakticky tak navzdory názvu řadí tento koncept na informačně-datovou úroveň. Podrobněji se analýzou definic Big Data zabývá (De Mauro et al. 2016). (Lněnička 2015) chápe tento pojem jako „sadu pokročilých technologií navržených k efektivnímu využití a údržbě dat, která nejsou jen velká, ale také rozmanitá a rychle se měnící".

Druhým souvisejícím pojmem **je datová integrace**. Tu podrobněji člení publikace (Vodňanský 2016b) na interní (provazující samotné datové zdroje a tím i jimi reprezentovanou informaci) a externí (propojující vrstvu dat s ostatními vrstvami podnikové nebo organizační informatiky, zejména procesy, potažmo i s okolním světem). Jedním z hlavních fenoménů externí datové integrace na celosvětové úrovni je pak koncept Linked Data a sémantického webu, který popisují (Svátek a Vacura 2007; Heath a Bizer 2011) a který v souladu se svým cílem postupně utváří jednotný globální datový prostor. Dále pak tato

integrace může být chápána jako proces (průběžný či jednorázový, související s data governance, popsanou (Begg a Caira 2012; Zach et al. 2014)) nebo jako kvalitativní stav (míra, do niž data jsou v jistý okamžik integrována).

Tato kapitola se v prvé řadě zabývá formalizovaným popisem výpočtu hodnot tří metrik na datových objektech. Poté je definován objekt v kontextu datové úrovně. V dalších částech jsou vymezeny jednotlivé **formáty** dat, případně **datové modely**, které jsou v současnosti typicky využívány a tato práce je bere v potaz. Na základě navržené podoby metrik je stanoveno, jakého rozsahu hodnot metrik nabývají data odpovídající datovému formátu či modelu v ideálním případě.

6.1 Relační databáze

Relační databáze (RDBS) v kontextu této práce je chápána coby relační datový model bez vazby na konkrétní produkt či implementaci.

V oblasti strukturovaných dat jsou jednotlivé implementace relačních databází zcela bezpochyby dominantní technologií, především pak v podnikové oblasti, jak potvrzuje (Ramel 2015). Tento velmi elegantní koncept známý na vědecké úrovni především na základě publikace (Codd 1990) dodnes v mnoha ohledech není překonán. I přes to se v kontextu této práce potýká s řadou **nepružností**, které systém metrik pomáhá podrobněji specifikovat.

Prvním takovýmto problémem je zaměření na **strukturovaná data**. Tento problém, kdy RDBS není schopna nativně (spolu se zachováním konzistence, ačkoli jinak je možné uložit do databáze libovolně strukturovaná data) uložit semistrukturovaná až nestrukturovaná data, popisuje (Florescu 2005). Existuje řada technologií, které tento problém více či méně řeší, například:

- podporou datového typu XML (Microsoft 2017),
- podporou speciálních XML úložišť v rámci RDBS (Oracle nedatováno),
- podporou JSON a XML coby datových typů sloupce (typické například pro PostgreSQL) a
- nativními XML databázemi, jejichž datový model navrhl již například (Wuwongse et al. 2003).

Dalším významným aspektem RDBS je její jistá **uzavřenost**. Ačkoli její samotná data mohou být zpřístupněna odkudkoli, jejich relace jsou formalizovány pouze uvnitř databázového systému a jejich propojení vně systému není v jejich tradičním pojetí podporováno.

S tím souvisí i **problémy se škálovatelností** a nízkou flexibilitou klasických RDBS, jejichž manipulace s daty pro svou častou vyšší náročnost přinášejí nutnost cacheování (ukládání do mezipaměti) přístupů do databáze, což navzdory zrychlení velmi komplikuje možnosti zajištění konzistence, neboť zde vzniká redundance. Tyto a obdobné problémy se snaží řešit relativně nový koncept NoSQL[4] databáze coby reakce na stále narůstající objemy dat. Tento koncept ukládá data v podobě položek typu „klíč -> hodnota“, přičemž dle (Madison et al. 2015) je kvůli zvýšení výkonnosti obětován koncept ACID[5] chápaný v RDBS coby defacto norma. Naopak v oblasti NoSQL databází v současnosti neexistuje významný standard (jako je tomu i v případě jazyka SQL) a je tak velice obtížné tento neustálený koncept odborně zkoumat. Řada jeho pojetí navíc umožňuje ukládání ve formátech zkoumaných dalšími sekcemi, jako je například a zejména formát JSON.

Dále je podstatným aspektem relačních struktur jejich **normalizace**. Podobně jako u strukturovanosti i zde platí, že optimálně navržená databáze obsahuje data s **minimem redundance** (ačkoli prakticky je možné opět uložit data libovolně redundantní).

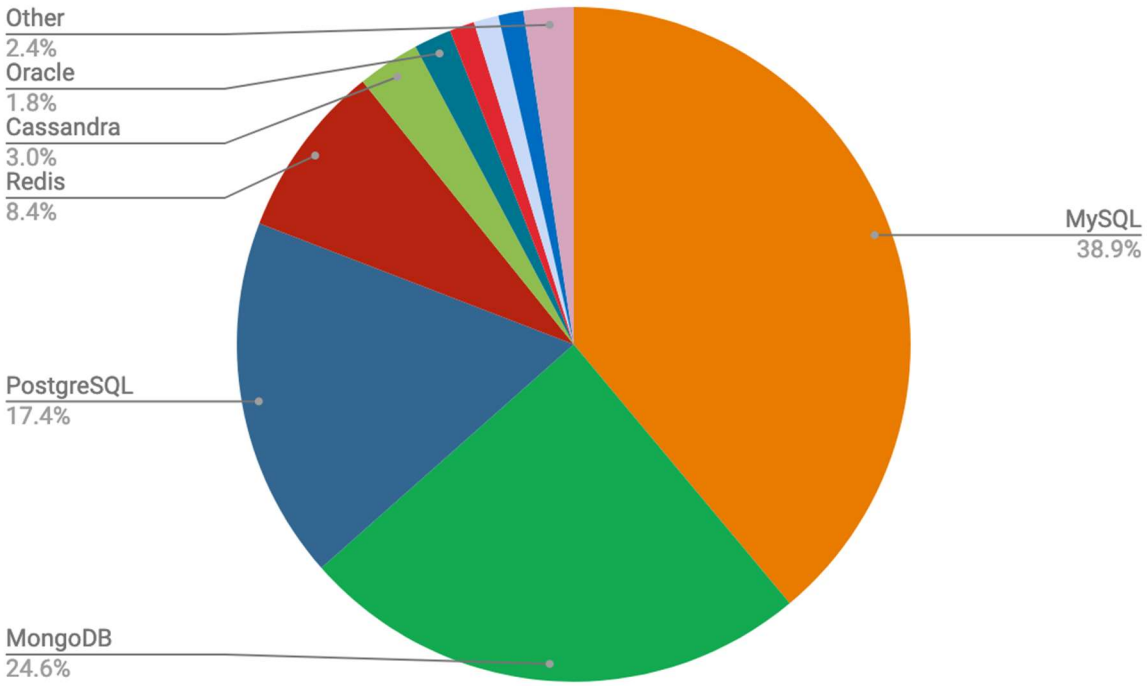
Celkově značnou dominanci podílu relačních databází mezi ostatními typy úložišť (vyjma MongoDB a částečně Cassandra nebo ElasticSearch) potvrzují výzkumy dle (Anderson 2019; Explore Group 2019; Ramel 2019; solid IT 2019).

Současný stav využití databázových technologií (v oblasti open-source, což ale pro srovnání podílů SQL a NoSQL není příliš podstatné) znázorňuje Obrázek 9. Dále (Scalegrid.io

[4] „No“ v tomto ohledu nepředstavuje anglické „ne“, nýbrž spojení „not only“ a jedná se tak spíše o rozšíření klasického SQL konceptu

[5] Koncept ACID - Atomicity, Consistency, Isolation, Durability – vlastnosti, které by měla zachovat každá databázová transakce

2019) uvádí, že celkový podíl NoSQL technologií oproti SQL je 39,5% a nejtypičtější kombinací použití dvou technologií je MySQL + MongoDB (34,15%), přičemž celkově až 75,6% respondentů kombinuje SQL a NoSQL technologii. Z hlediska výpočtu hodnot metrik a celkového kontextu této práce lze usoudit, že NoSQL databáze fakticky pouze umožňují strukturované uložení odpovídající formátu JSON nebo XML (popřípadě příbuzného formátu s prakticky shodnými vlastnostmi), kterým je věnována část 6.2. I vysoký podíl kombinace SQL a NoSQL technologií dokládá, že NoSQL patrně není doposud ideálním řešením omezení klasických relačních databází uvedených na začátku této části.

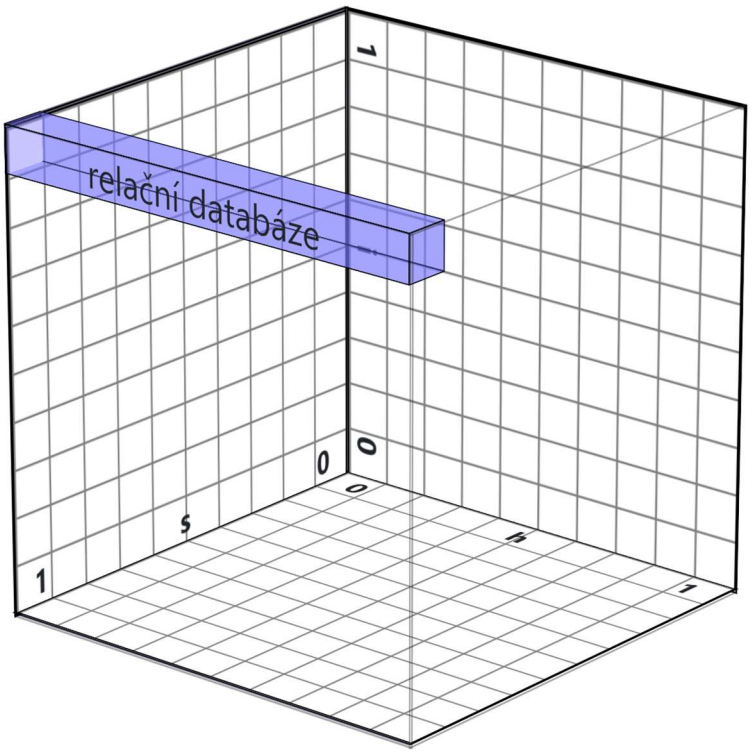


Obrázek 9 – podíl využití open-source databázových technologií v roce 2019 (Scalegrid.io 2019)

Obrázek 10 znázorňuje prostor, který vymezuje oblast pokrývanou daty na základě zkoumaných metrik ideálně navržené RDBS, tedy míra informace (i) a míra strukturovanosti (s) jsou rovny jedné (RDBS jsou určeny pro co nejvíce normalizovaná a strukturovaná data), zatímco míra hierarchičnosti může být libovolná, a proto $h \in \langle 0; 1 \rangle$. Pro přehlednost jsou hodnoty škálovány do intervalů o velikosti 0,1. Jak je již uvedeno, prakticky je umožněna libovolná redundance i strukturovanost, což je ovšem v rozporu s pojetím RDBS již od jejího původního konceptu (Codd 1990) a kvádr v dotyčné vizualizaci by obsáhl celý znázorněný krychlový prostor.

Datový objektem relační databáze je typicky

- položka (buňka),
- řádek tabulky,
- tabulka,
- databáze (nebo schéma, záleží na konkrétním systému) a
- celý databázový systém.



Obrázek 10 – vizualizace trojrozměrného metadatového prostoru ideálně navržené RDBS

6.2 XML a JSON

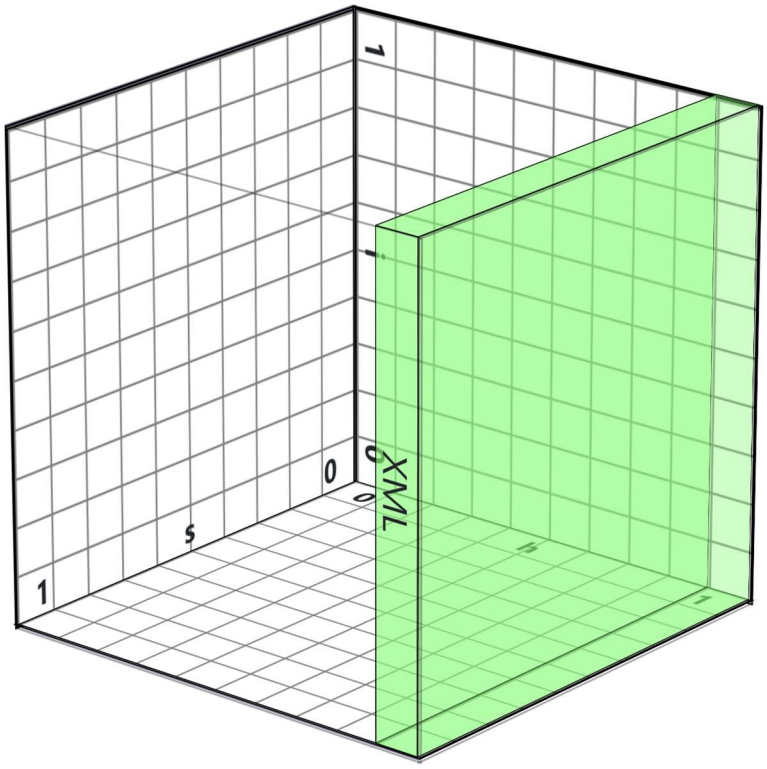
Tyto dva formáty jsou pro svou podobnost zkoumány společně. Tato podobnost spočívá především v čistě **hierarchické struktuře** a typickém použití ve webových technologiích a datové výměně. Příbuznost těchto formátů potvrzuje i (Helland 2017) zmiňující především jejich flexibilitu a sebepopisnost v porovnání vůči relačním databázím.

Zde je nutné podotknout, že XML je hierarchickým formátem pouze ve svém **tradičním pojetí,** tj. ve chvíli, kdy relace jsou vyjádřeny pouze na základě vztahů elementů rodiče nebo potomka. Tímto způsobem v praxi nemusí být a často není XML používáno; jazyky schémat XML umožňují různými způsoby užívat obdoby cizích klíčů v relační databázi, které vytváří relace napříč hierarchií. Tento přístup, ačkoli umožňuje kontrolu referenční

integrity, je však do jisté míry nekoncepční, neboť reprezentuje **relace dvojím způsobem** a může tak umožnit modelovat jednu informaci mnoha různými způsoby. Dále pak jsou tyto obdoby cizích klíčů užívány pro XML prakticky neformálním způsobem, jako např. zmíněná serializace RDF/XML, která veškeré relace staví pouze na URI identifikátorech, navíc bez kontroly konzistence. O tomto rozporu dále mluví část 6.3.

Zde lze XML dokumenty dle (Nečaský et al. 2008) rozdělit na

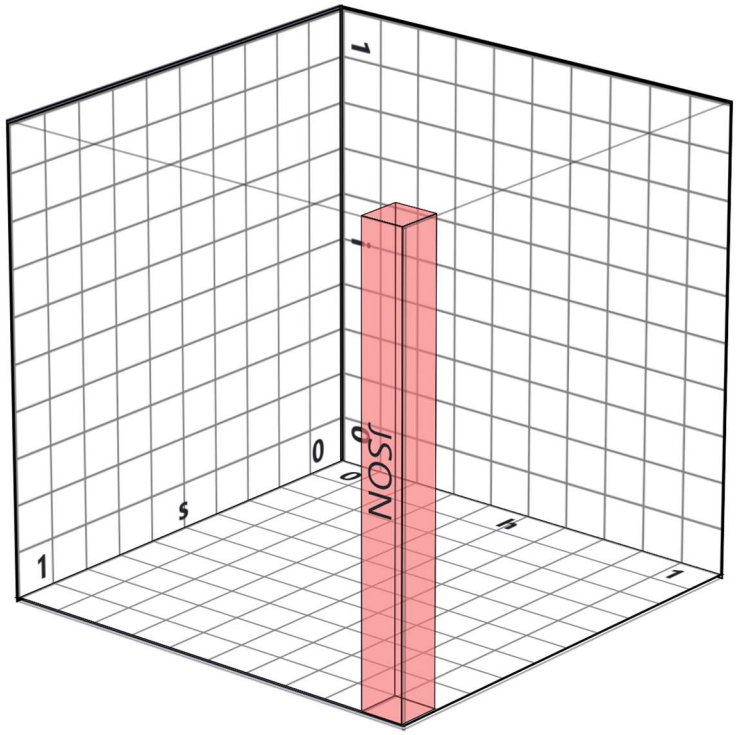
- **datově orientované** – vytvářené a zpracovávané zpravidla strojově, bez podstatného pořadí elementů a strukturované,
- **dokumentově orientované** – vytvářené a zpracovávané (čtené) přinejmenším zčásti člověkem, semistrukturované a s podstatným pořadím elementů a
- **hybridní dokumenty**, kombinující oba přístupy.



Obrázek 11 – vizualizace trojrozměrného metadatového prostoru formátu XML

Obrázek 11 znázorňuje prostor, který na základě zkoumaných metrik obsáhnou v ideálním případě data formátu XML. Hodnoty jsou stejně jako v předchozí vizualizaci pro přehlednost škálovány do deseti intervalů o velikosti 0,1, přičemž míra informace (i) a míry strukturovanosti (s) jsou libovolné. Na XML se totiž nevážou standardizované metody normalizace dat, ačkoli jejich návrhy existují (Arenas a Libkin 2004). Míra hierarchičnosti (h) je rovna jedné, jak zdůvodnil první odstavec této sekce.

Formát JSON (JavaScript Object Notation) je pro svou úspornost zápisu (byť částečně na úkor přehlednosti pro člověka) v poslední době populární zejména v metodách webových aplikací. S XML se do značné míry překrývá, jak je patrné z vizualizací (Obrázek 11 a Obrázek 12).



Obrázek 12 - vizualizace trojrozměrného metadatového prostoru formátu JSON

Na rozdíl od XML ovšem nativně nepodporuje semistrukturovaná a nestrukturovaná data (byť jejich samotné uložení opět je možné) a vzhledem k tomu, že vznikl coby zápis objektu programovacího jazyka, nemá své schéma (ačkoli programátorsky lze jeho strukturu hlídat např. použitím rozhraní).

Datovým objektem v XML je zejména

- element, případně i
- jeho obsah,
- hodnoty atributů a

doplňující specifikace, má-li smysl je zkoumat odděleně. V JSONu se jedná o

- objekty,
- pole,
- klíče a
- koncové atomické hodnoty (elementární datové objekty).

Metadatový prostor pokrytý formáty XML a JSON se nevztahuje pouze na samotné dokumenty uložené v tomto formátu. Tentýž princip lze vztáhnout na odvozené technologie, jakými jsou XML databáze, popřípadě databáze konceptu NoSQL a key-value, jakými jsou MongoDB a Redis.

6.3 RDF, RDFS a OWL

Ontologické jazyky, o kterých mluví tato sekce (RDF, RDFS a OWL), mají při shodné sémantice (představující do jisté míry informační úroveň modelování) řadu tzv. serializací, tedy způsobů textové datové reprezentace. Jednou z těchto hlavních serializací je formát založený na XML, ačkoli z hlediska datových metrik nemá s pojetím XML z předchozí části mnoho společného.

Jak již bylo zmíněno v předchozí části týkající se formátu XML, existuje zde rozpor mezi jeho tradičním pojetím a pojetím nahrazujícím klasické stromové relace za jistou z hlediska XML neformální obdobu užití cizích klíčů známých z relačních databází. Tento rozpor zmiňuje i (Svátek 2002, s. 16), zmiňující výhodu „grafového modelu“ RDF oproti „gramatickému“, tedy stromovému XML, spočívající v „otevřenosti pro vyjádření sémantiky metadat prostřednictvím ontologií“, kdy je možné vytvářet zcela nezávislá jednotlivá tvrzení propojitelná s ostatními.

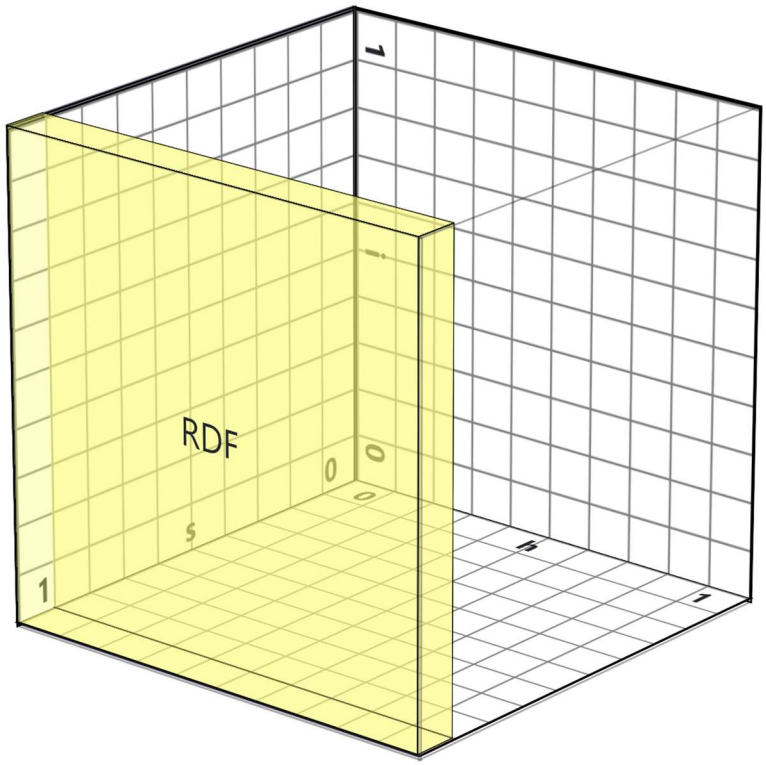
Obecně lze členit tyto jazyky na úroveň A-Boxu, jemuž odpovídá formát RDF a úroveň T-Boxu, do nějž spadá RDFS a další konstrukty přidávající OWL. Jedním z hlavních účelů těchto jazyků je koncept sémantického webu, umožňujícího strojové čtení veřejně dostupného obsahu přímo na informační úrovni, a tedy s možností automatizace inferenčních procesů.

Vzhledem k tomu, že se jedná fakticky v případě webu o jeden celosvětový a decentralizovaný datový prostor, označovaný také jako „propojená data“, mohou být tato data zcela **libovolně redundantní**, tedy i nekonzistentní a lze předpokládat, že do značné míry také redundantní jsou. V důsledku toho lze výroky uspořádané do zmíněných trojic pokládat za de-facto čtveřice, kde čtvrtým atributem je dotyčnou trojici publikující subjekt – zde

může jít o název domény, popřípadě na ní navázaný SPARQL endpoint, umožňující dotazování do těchto dat nebo IRI pojmenovaného grafu. Pomocí něj lze vyhodnocovat pravdivost konkrétních sdělení v případě přímých konfliktů nebo nesrovnalostí, což lze pokládat za námět pro další výzkumy, které nebyly k tomuto tématu doposud dohledány. O nekonzistenci ontologických dat mluví i (Svátek 2002, s. 26–27).

Data odpovídající RDF lze tedy pokládat v nejtypičtějším případě za **zcela strukturovaná, libovolně hierarchická a libovolně redundantní**, jak zachycuje Obrázek 13.

Datovým objektem je v případě RDF (je-li uvažována opět úroveň A-Boxu) URI identifikátor.



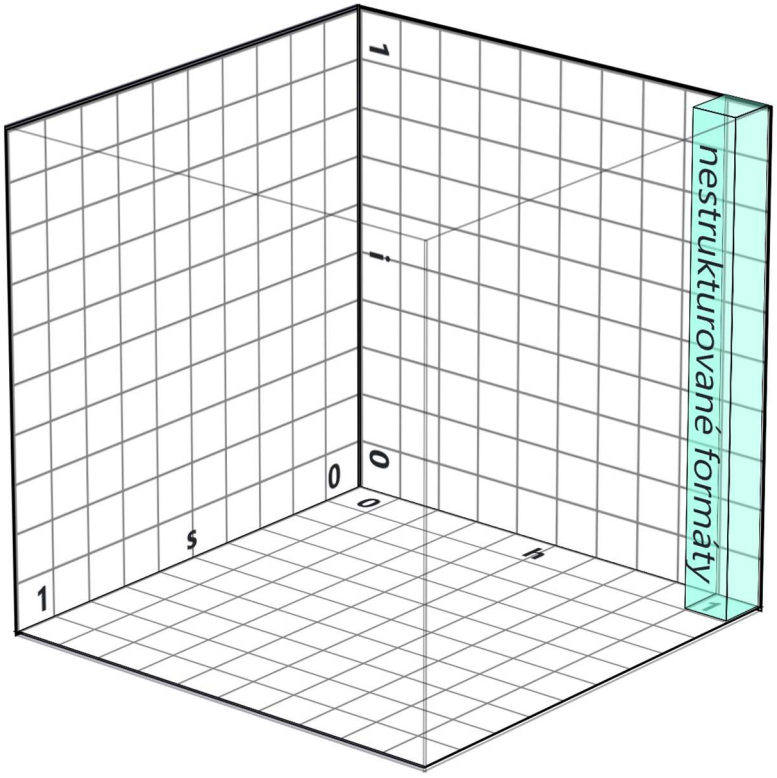
Obrázek 13 - vizualizace trojrozměrného metadatového prostoru RDF

6.4 Nestrukturované formáty

Nestrukturovaným formátem dat lze myslet každý formát, který neobsahuje nativní (přímo strojově čitelnou) strukturu, bez ohledu na strukturovanost jím reprezentované informace. Přímou strojovou čitelností je myšleno čtení formátu v jeho původní podobě bez užití inteligentních strukturalizačních metod, o kterých dále mluví sekce 7.2.1.

Do této skupiny lze jednoznačně zařadit zejména volné texty, obsah multimediálních souborů, ovšem (Beach a Schiefelbein 2014) říkají, že celkově jde o nejednoznačně vymezený pojem, kam často bývají řazena i data obsahující částečné struktury (fakticky semistrukturovaná), jako jsou e-maily. Nestrukturovaná data mohou být součástí i dat jinak strukturovaných, popřípadě majících strukturované schéma, tedy nejtypičtěji RDBS a XML, kde vzhledem k nestrukturovanosti na informační úrovni nelze tento jev vhodným způsobem eliminovat. V kontextu této práce nestrukturovaná data jsou **zcela hierarchická** (absence vnitřní explicitní struktury znemožňuje výskyt porušení definice hierarchičnosti - Vzorec 1). Míru informace u nich lze prakticky počítat pouze na základě AIT, tedy prostřednictvím komprese. Z toho důvodu je nutné tato data chápat jako atomický (pro účely výpočtu metrik dále nedělitelný a vnitřně nezkoumatelný), tedy elementární datový objekt v souladu s definicí v části 3.1.

Vizualizace nestrukturovaných dat je do značné míry sporná vzhledem k diverzitě jednotlivých formátů, jejichž zařazení do této skupiny navíc není vždy jednoznačné, v kontextu této práce ovšem veškerá takováto data jsou chápána způsobem, jaký znázorňuje Obrázek 14.



Obrázek 14 – obecná vizualizace trojrozměrného metadatového prostoru nestrukturovaných datových formátů

Jeden konkrétní nestrukturovaný datový objekt je tedy z hlediska výpočtu vždy sám o sobě atomický, tedy elementární.

6.5 Srovnání typů datových struktur

Systém tří metrik dle kapitoly 5 umožňuje přehledným způsobem klasifikovat jednotlivé typy datových struktur zmíněné v kapitole 6.

	Míra strukturovanosti	Míra hierarchičnosti	Míra informace
Relační databáze	{1}	⟨0; 1⟩	{1}
XML	⟨0; 1⟩	{1}	(0; 1⟩
JSON	{1}	{1}	(0; 1⟩
RDF	{1}	⟨0; 1⟩	(0; 1⟩
Nestrukturované formáty	{0}	{1}	(0; 1⟩

Tabulka 2 – porovnání ideálních hodnot metrik dat odpovídajících jednotlivým typům datových struktur

Tabulka 2 shrnuje jednotlivé hodnoty. Důležité je, že krajní hodnoty představují pouze **ideální stav** dat zcela kompatibilních s daným formátem nebo typem datového modelu a nikoli nutnou podmínku pro jejich uložení. Míra informace je myšlena coby inverzní pojem k redundanci, jak bylo zdůvodněno v sekci 5.4. Hierarchičnost nestrukturovaných formátů je dle definice v sekci 4.2.2 stanovena jako hodnota 1. Dále je možné, že metriky dokážou tyto jednotlivé a četně používané typy datových struktur **odlišit**, neboť kombinace hodnot v tabulce se pro žádnou jejich dvojici neopakují.

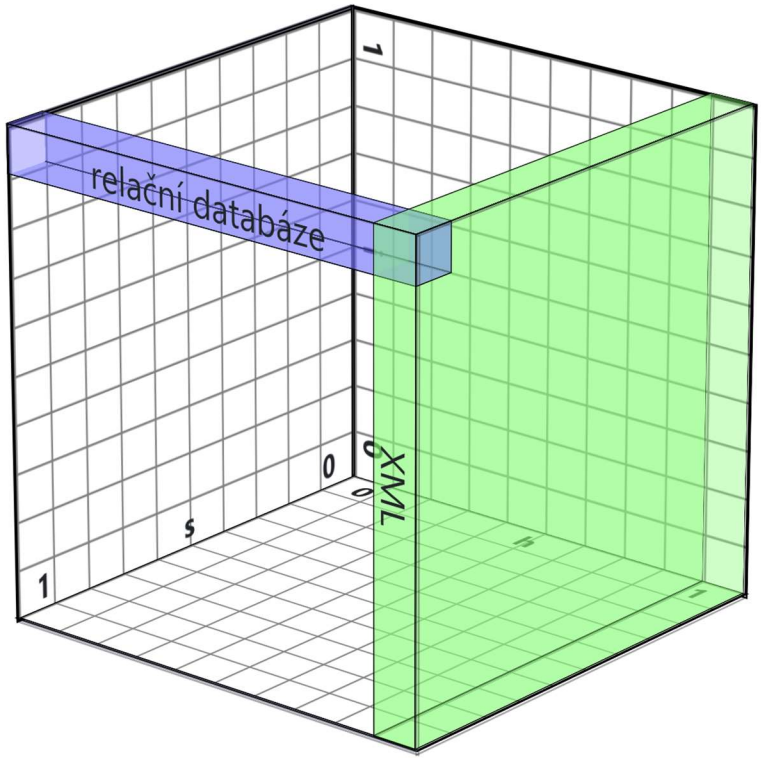
Toto členění odpovídá **vizualizacím** dotyčných struktur v předchozích částech této práce (kde jsou používány intervaly o velikosti 0,1, jak již bylo vysvětleno).

6.6 Analýza průniků

Průniky dvou i více geometrických reprezentací typů datových struktur umožňují analyzovat jejich kompatibilitu – geometrický útvar průniku odpovídající vymezuje oblast dat

zcela kompatibilních s oběma typy datových struktur. Datové objekty náležící průniku v tomto případě pak značí data přímo kompatibilní, a tedy převoditelná mezi oběma objekty či datovými strukturami bez informační ztráty.

Průnik typů datových struktur ilustruje například Obrázek 15, z něhož je vizuálně zřejmé, že s oběma typy datových struktur jsou zcela kompatibilní strukturovaná, hierarchická a normalizovaná data.



Obrázek 15 – příklad průniku pokryté oblasti RDBS a XML v trojrozměrném metadatovém prostoru

7 Datové transformace

Následující kapitola se zabývá dynamickou změnou metrik v čase, tedy posunem mezi dvěma (případě i více) stavy, a to pro jeden datový objekt. Datová transformace předpokládá, že nedochází ke změnám na úrovni informace a ve výchozím i cílovém stavu tak daty reprezentovaná informace je zachována.

Transformacemi na úrovni informace se tato práce nezabývá ze stejných důvodů jako jsou uvedeny v části 4.2 – nelze vypočítat hodnotu metriky pro informaci samotnou, neboť je vždy pracováno pouze s její více či méně přesnou reprezentací. V důsledku toho informační transformace spočívá ve změnách stavu samotné reality (bez ohledu na to, zda je jakkoli sdělována), což i mimo jiné přesahuje rámec této práce a samotného oboru informatika.

7.1 Míra kompatibility

Kompatibilita jednotlivých informačních a datových struktur, popřípadě pak jejich převoditelnost (určitá snadnost transformace jedné struktury do struktury odlišné) může být rozčleněna mimo již zmíněné osy data/informace právě na základě metrik navržených v částech 4.2 a 5.

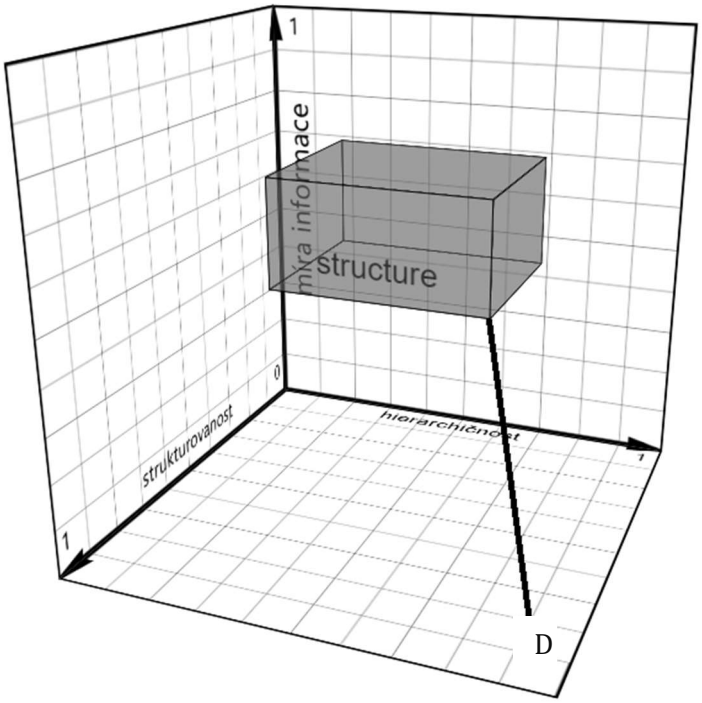
Kompatibilita konkrétních dvojic datových struktur je řešena celou řadou dřívějších výzkumů; kromě pravděpodobně nejtypičtějšího srovnávání RDBS a XML dále například (Aasman 2006) navrhuje sjednotit schéma RDBS a RDF triples (trojic) vlastním návrhem databáze určené přímo pro uložení RDF. Zde je ovšem sporné, zda ukládání trojic je v souladu s obecnými principy návrhu RDBS a její normalizace.

Míra kompatibility dvou datových objektů je vyjádřitelná celkovou délkou jejich transformace, jakou detailně popisuje Vzorec 13 v části 7.3. Tento výpočet ovšem pro zjišťování kompatibility nepostačuje; kompatibilita je obvykle zjišťována za účelem návrhu datové transformace a součástí tohoto procesu je i výběr nejvhodnější cílové datové struktury. Abstrahujeme-li tak od ostatních okolností dotyčné transformace (například je-li jejím primárním cílem některý ze směrů transformace uvedený v následující části), je cílem, aby délka transformace byla minimální, tedy nejvíce odpovídající hodnotám metrik dat. Délka transformace pak může sloužit k porovnání navržené transformace na základě změřené míry kompatibility s jejím reálným výsledkem.

Kompatibilita datového objektu D s typem datové struktury T, který je dán množinou hodnot metrik jemu odpovídajících datových objektů, může být spočítána na základě vzdálenosti vektoru objektu od intervalů nebo hodnot metrik pro dotyčný typ struktury, jaké uvádí Tabulka 2.

$$k_T(D) = 1 - \frac{1}{\sqrt{3}} \cdot \min_{S \in T} \sqrt{\big(s(S) - s(D)\big)^2 + \big(i(S) - i(D)\big)^2 + \big(h(S) - h(D)\big)^2}$$

Vzorec 10 – výpočet kompatibility datového objektu D a typu datové struktury T



Obrázek 16 – princip výpočtu míry kompatibility na základě vzdálenosti datového objektu od typu datové struktury

Tento způsob výpočtu formalizuje Vzorec 10. Maximální možná vzdálenost (nejnižší kompatibilita) odpovídá tělesové úhlopříčce krychle o délce $\sqrt{3}$ – takovýchto hypotetických transformací je tedy osm (čtyři tělesové úhlopříčky ve dvou směrech). Funkce $k_T(D)$ je mírou obecné kompatibility datového objektu D, kde minimum se počítá přes všechny datové objekty odpovídající danému typu datové struktury T. Geometricky tak hodnota kompatibility je založena na vzdálenosti bodu a kvádru (potažmo i obdélníku, úsečky či dalšího bodu, pokud typ datové struktury u některé z metrik je charakterizován nikoli intervalem, ale číslem) v trojrozměrném metadatovém prostoru, což znázorňuje Obrázek

16. Dále je ovšem hodnota míry kompatibility škálována na interval ⟨0; 1⟩, přičemž 0 značí zcela nekompatibilní a 1 zcela kompatibilní dvojici.

Z tohoto vzorce lze odvodit specifické výpočty kompatibility datových objektů s konkrétními čtyřmi typy datových struktur, a to následujícími způsoby:

$$k_{RDBS}(D) = 1 - \frac{1}{\sqrt{3}} \cdot \sqrt{\big(1 - s(D)\big)^2 + \big(1 - i(D)\big)^2}$$

$$k_{XML}(D) = 1 - \frac{1}{\sqrt{3}} \cdot \big(1 - h(D)\big)$$

$$k_{JSON}(D) = 1 - \frac{1}{\sqrt{3}} \cdot \sqrt{\big(1 - s(D)\big)^2 + \big(1 - h(D)\big)^2}$$

$$k_{RDF}(D) = 1 - \frac{1}{\sqrt{3}} \cdot \big(1 - s(D)\big)$$

Vzorec 11 – výpočet kompatibility datového objektu D a specifických typů datových struktur

7.2 Analýza možných výchozích a cílových struktur a směrů transformace

Tato část prezentuje základní a nejjednodušší příklady transformace dat. Systém tří metrik umožňuje klasifikovat existující datové transformace. Tyto transformace se mohou odehrávat ve dvou směrech, kdy dotyčná hodnota metriky narůstá (případně, avšak ne nutně ze své minimální na svou maximální hodnotu) nebo naopak klesá (v tomto případě k základnímu pojmu je doplněna předpona „de-“).

Strukturalizace	vzestup míry strukturovanosti	**S**
Destrukturalizace	pokles míry strukturovanosti	**DS**
Hierarchizace	vzestup míry hierarchičnosti	**H**
Dehierarchizace	pokles míry hierarchičnosti	**DH**
Normalizace	vzestup míry informace	**N**
Denormalizace	pokles míry informace	**DN**

Tabulka 3 – základní směry transformace a jejich značení

Dotyčná transformace může mít ve všech případech, popřípadě jen některých konkrétních, jistý pozitivní nebo negativní vliv i na ostatní metriky, čímž dochází ke kombinovaným transformacím, jimiž se zabývá část 7.3.

7.2.1 Strukturalizace

Procesem strukturalizace je myšlena **transformace datového objektu o určité míře strukturovanosti do objektu o míře strukturovanosti vyšší až maximální**, tedy rovné jedné. Jak již bylo upozorněno v sekci 4.2.1, může spolu s tímto procesem dojít i k jistým informačním ztrátám (především v případu nestrukturovaného textu). K těm dochází, pokud samotná struktura (uspořádání slov a vět) má sémantický význam, který může spočívat například i v pouhé čtivosti (snadnosti dekódování) pro lidský mozek, popřípadě pak v další informaci spíše emočního a podvědomého charakteru. Proces strukturalizace je proveditelný člověkem a v praxi často prováděný v řadě profesí (typickým příkladem je **účetnictví**, nebo také metody **modelování** při návrhu systému).

Automatizace strukturalizace vyžaduje jistou míru strojového pochopení sémantiky. Do jisté míry se mu blíží pojem „sémantická anotace“, jehož jednoznačná obecně uznávaná definice pravděpodobně neexistuje. Vysvětlením tohoto pojmu se zabývá (Oren et al. 2006), který zmiňuje zejména metody značkující pojmy v nestrukturovaném textu za účelem jejich vložení do relací (konkrétně zmiňuje doplňování odkazů na webu). Takovýto postup prakticky znamená transformaci nestrukturovaných dat do semistrukturovaných, tedy jistou neúplnou podmnožinu celého procesu strukturalizace.

Jedním z nejtypičtějších příkladů plně automatizované absolutní strukturalizace jsou nástroje převádějící věty **přirozeného jazyka** (popřípadě i jeho semistrukturované podoby

například ve formátu HTML) do formátů RDF/OWL (nebo i případné jiné podobné nástroje s jiným cílovým strukturovaným formátem). Příkladem je nástroj FRED, generující tímto způsobem zápis v řadě RDF serializací i grafovou podobu (Gangemi et al. 2017). Dalším příkladem je **technologie OCR**[6] rozpoznávající písmo z obrazového formátu (zpravidla bitmapová grafika, zde nejde o absolutní strukturalizaci, pouze o zvýšení míry strukturovanosti) nebo technologie **rozpoznávání slov** ve zvukovém záznamu mluvené řeči. Jako příklad strukturalizace lze brát i „aktivity třídění, čištění a transformace nestrukturovaných dat do matic", které zmiňuje (Šperková 2014) coby náročný proces při analýze dat ze sociálních sítí.

Proces strukturalizace může mít vliv i na normalizaci dat, zejména v případech, jsou-li transformována do strojově čitelné podoby, popřípadě může normalizací zároveň být. Míra hierarchičnosti může v takovémto případě klesat. Lze si ovšem představit i příklady, kdy míra hierarchičnosti naopak narůstá, pokud hierarchický formát je vyžadován okolnostmi, např. formáty typu XML. V tomto případě k normalizaci obvykle nedochází, naopak je možná denormalizace.

Strukturalizace má tedy **neutrální vztah** ke všem čtyřem ostatním typům transformací (tedy na ostatních osách v trojrozměrném datovém protoru), neboť umožňuje jejich paralelní vzestup i pokles. Nemusí ho ovšem umožňovat v rámci trojitých transformací, měnících zároveň i hodnoty míry hierarchičnosti a míry informace, jejichž vztah zkoumají následující sekce).

7.2.2 Destrukturalizace

Inverzní proces vůči strukturalizaci lze naopak pokládat za velice prostý a běžný v každodenní (zejména mezilidské) komunikaci. Dochází k němu v případech, kdy původně strukturovaná data jsou interpretována **méně strukturovaným způsobem**, obvykle větami přirozeného jazyka. Nejčastěji se tak patrně děje za účelem vyšší míry srozumitelnosti pro člověka. Typickým příkladem je prostá prezentace informace reprezentované původně více strukturovanými daty vůči člověku (coby fyzický akt, nikoli dokument).

Proces destrukturalizace opačně vůči strukturalizaci může sám o sobě naopak vytvářet přidanou informaci o charakteru shodném jako v předchozí sekci – tu ovšem nelze uvažovat, neboť jak stanovila úvodní podmínka v části 7.2, transformace na úrovni informace

[6] Optical Character Recognition

jsou vyloučeny. Ke vzniku dodatečné informace dochází zejména v případě, kdy je proces prováděn člověkem, který může být ovlivněn svými postoji, názory, emocemi i jazykovými zvyklostmi.

Destrukturalizace má vliv na míru hierarchičnosti, ovšem poměrně neobvyklým způsobem – znemožňuje její zkoumání a ruší i její samotný smysl, neboť hierarchičnost je tím méně zkoumatelná, čím méně jsou data strukturovaná. Při striktním pojetí definice hierarchické struktury ze sekce 5.3 nestrukturovaná data jsou sama o sobě elementární datový objekt – přestávají být dále dělitelná a případná porušení hierarchičnosti tak mizí; **míra hierarchičnosti** proto může pouze být **zachována nebo stoupat**. Vznik porušení hierarchičnosti – dehierarchizace – je tedy ze stejného důvodu vyloučena. Zároveň spolu s destrukturalizací může velmi pravděpodobně docházet k denormalizaci (u nestrukturovaných dat je obtížné až nemožné vyloučit redundance); s normalizací se destrukturalizace zcela nevylučuje – i méně strukturované vyjádření může teoreticky být méně redundantní.

Destrukturalizace má možný **pozitivní vliv na míru hierarchičnosti;** může probíhat s hierarchizací, **vylučuje dehierarchizaci**. Nestrukturovaná data jsou zcela hierarchická. Destrukturalizace může mít **negativní vliv na míru informace** – umožňuje denormalizaci, teoreticky ale ani **nevylučuje normalizaci**.

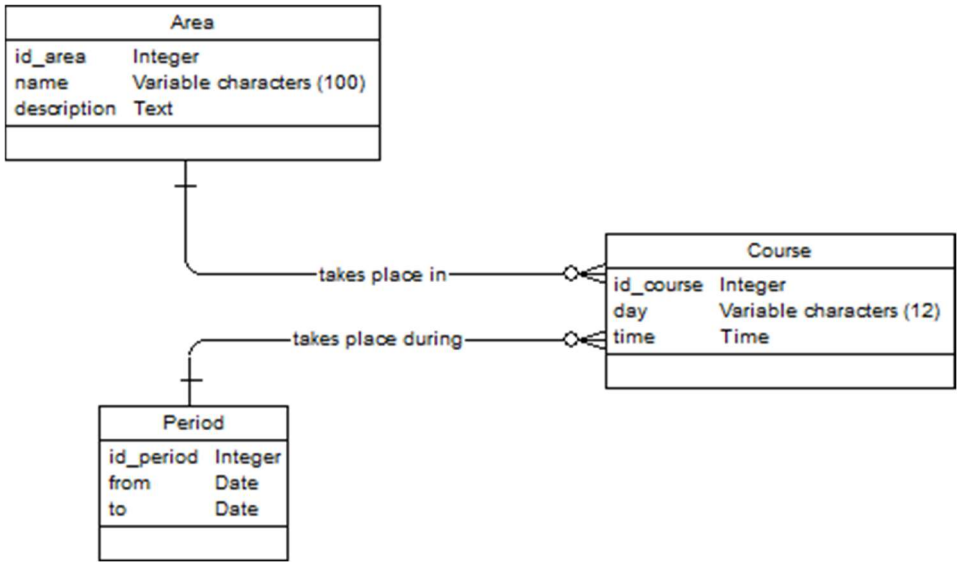
7.2.3 Hierarchizace

Procesem hierarchizace je myšlena **transformace datového objektu o určité míře hierarchičnosti do více či zcela hierarchického objektu**, mající míru hierarchičnosti zvýšenou oproti předchozímu stavu či rovnu jedné. Jak ukázala sekce 4.2.2, modelovaná realita ve skutečnosti jen velmi málokdy má zcela hierarchickou podstatu. Protože bývá modelována z jistého úhlu pohledu, bývá obvykle z naprosté většiny hierarchická, jak zkoumala dále kapitola 9 v oblasti ontologických slovníků. Je každopádně zřejmé, že dochází-li k hierarchizaci ne zcela hierarchických dat (tedy obsahujících strukturu porušující definici hierarchičnosti, kterou stanovuje Vzorec 1), je výsledná struktura neoptimální – obsahuje redundance.

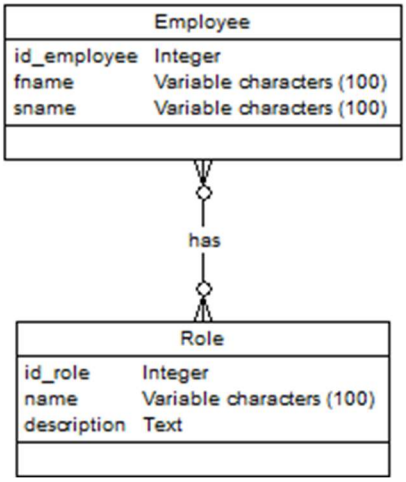
Hierarchizace je po praktické stránce velmi běžným procesem již z důvodů uvedených v sekci 4.2.2. – jedná se například o tvorbu naprosté většiny dokumentů včetně psaní této práce, jejíž struktura je hierarchická, ačkoli je dále provázaná křížovými odkazy na jednotlivé části, čímž je hierarchičnost porušena podobným způsobem jako obdoby cizích

klíčů v XML, o nichž mluví část 6.2. Dalším typickým příkladem hierarchizačního procesu je návrh webové stránky používající data z relační databáze. Obecně tedy jde o proces, v němž optimálnost uložení (nízká míra redundance a nízká velikost uložených dat) struktury z hlediska úložiště dat je nahrazována svou protiváhou. Tou může být zpracovatelnost pro lidský mozek (přehledností), případně i snazší, a tedy rychlejší dohledatelnost dat ve stromové struktuře.

Podstatou neoptimálnosti hierarchizované struktury se podrobněji zabývá (Vodňanský 2016a), uvádějící dva nejjednodušší možné příklady nehierarchické substruktury, prezentované na konceptuálním datovém modelu prostřednictvím ER diagramu, jak ukazuje Obrázek 17 a Obrázek 18.



Obrázek 17 – První příklad porušení definice hierarchičnosti - sbíhající hierarchie dle (Vodňanský 2016a)



Obrázek 18 – Druhý příklad porušení definice hierarchičnosti – vztah o kardinalitě $M:N$ dle (Vodňanský 2016a)

První obrázek znázorňuje schéma sportovní firmy, pořádající kurzy. Každý kurz se váže na areál a období, do nichž je vypsán. Druhý obrázek znázorňuje velmi jednoduché organizační schéma malé firmy, kde jsou rozlišováni pouze zaměstnanci a jejich role ve vztahu o kardinalitě $M:N$. Podstatou problému je, že obě struktury umožňují transformaci do dvou různých hierarchických struktur. V prvním případě je možné vytvořit hierarchii Areál → Období → Kurz, která může způsobit opakované vypisování jednotlivých období pro každý areál či opačnou hierarchii Období → Areál → Kurz, kde mohou redundantně být opakovány areály pro každé období. V druhém případě je možné vytvořit hierarchii Zaměstnanec → Role nebo opačnou Role → Zaměstnanec s analogickými problémy.

Na základě pouhé znalosti T-Boxu není možné určit, které z možných transformovaných struktur vytvoří na úrovni dat menší redundanci. Na úrovni A-Boxu ovšem toto rozhodnutí možné je – informace navzdory svému schématu může být zcela hierarchická (v každém období jsou provozovány jiné areály, každý zaměstnanec má jen jednu roli apod.) což směr hierarchizace určí zcela jednoznačně. Popřípadě informace může mít míra hierarchičnost podstatně vyšší než její schéma, ačkoli ne rovnou jedné. V takovémto případě je nejvhodnějším řešením experimentální hierarchizace v obou možných směrech[7] a srovnání redundance struktur pomocí postupů uvedených v sekci 5.4.

Použitá hodnota, založená na principu entropie je poté nejsnáze vypočtena prostřednictvím nahrazení pravděpodobnosti četností opakování veškerých uzlů ve výsledné hierar-

[7] těchto směrů může v závislosti na kardinalitě entity porušující definici hierarchičnosti být i více, než dva

chizované struktuře, nejtypičtěji v XML dokumentu. Výsledné dvojice hodnot (předpokládáme-li srovnání pouze dvou hierarchizovaných struktur) rozlišuje (Vodňanský 2016a) na tři možné typy situací:

1. obě hodnoty vyjdou nízké[8] – hierarchizace bez vzniku výrazné redundance není možná a je vhodné zvážit účel dotyčné transformace,
2. hodnoty vyjdou výrazně odlišné (vysoká a nízká) – na úrovni A-boxu se struktura velmi blíží kardinalitě 1: *N* a optimální směr hierarchizace je zřetelně určen,
3. hodnoty vyjdou obě vysoké – na úrovni A-boxu se struktura velmi blíží kardinalitě 1: 1 a oba směry hierarchizace jsou podobně či stejně vhodné.

Srovnání na principu Shannonovy informační entropie má nevýhody uvedené již v sekci 5.4, zde zejména nemožnost výpočtu na komplexnějších strukturách nežli lineární sekvenci (skládající se v tomto případě z jednotlivých XML elementů). Lze ji proto obdobně nahradit již uvedenou mírou vycházející z AIT vypočtenou pomocí kompresního poměru.

Závěrem lze říct, že hierarchizace, tedy transformace ne zcela hierarchických dat reprezentujících ne zcela hierarchickou informaci je zároveň procesem pravděpodobně vytvářejícím redundanci (tedy denormalizace, naopak normalizace je vyloučena). Zároveň je však procesem zcela běžným a v mnoha situacích i nevyhnutelným, který je z uvedených příčin vhodné optimalizovat, aby vzniklá redundance byla minimální. V rámci hierarchizace přitom je možná jak strukturalizace dat (probíhá-li s omezením na hierarchickou strukturu a jedná-li se tak zároveň pravděpodobně o denormalizaci), tak i destrukturalizace, neboť nestrukturovaná data jsou hierarchická, jak uvedla již sekce 4.2.2.

Hierarchizace má tedy **negativní vliv na míru informace** (umožňuje denormalizaci, vylučuje normalizaci) a zcela **neutrální vztah ke změnám míry strukturovanosti**.

7.2.4 Dehierarchizace

Inverzní pojem značí proces, kdy definice hierarchické vlastnosti (Vzorec 1) je transformací porušena. Může se tak dít v rámci strukturalizačního procesu, neboť pouze jiná než nestrukturovaná data umožňují porušení hierarchie, jak uvedla i předchozí sekce. Dále k dehierarchizaci dochází v rámci normalizačního procesu, neboť datové hierarchické substruktury odporující schématu informace jsou jednou z příčin vzniku redundance.

[8] jaká hodnota je již nízká a jaká vysoká nelze přesně určit a závisí na konkrétní aktuální toleranci redundance

Čistá dehierarchizace tak patrně nemá svůj typický příklad a je obvykle součástí obou těchto zmíněných procesů, nicméně neexistuje důvod vyloučit její možnost.

Dehierarchizace má tedy **pozitivní vliv na míru informace** (hlavním důvodem pro narušování hierarchie je normalizační proces, ač teoreticky nemusí být podmínkou, ovšem denormalizaci spolu s dehierarchizací lze pokládat za vyloučenou). Také může mít **pozitivní vliv na strukturovanost** (destrukturalizace je vyloučena; propojováním dat nehierarchickým způsobem nelze snížit míru strukturovanosti).

7.2.5 Normalizace

V souladu s principy dle (Codd 1990; Halpin 2001) je procesem normalizace, resp. aplikace tzv. normálních forem myšlena **úprava schématu** relační databáze do podoby minimalizující možný vznik redundance a tedy i nekonzistentního stavu. Samotná data dle tohoto konceptu upravována nejsou. Normalizační proces se dle (Kent 1983) řídí zejména pěti návaznými pravidly, nazvaných normální formy. Tato problematika je široce a dlouhodobě zkoumána a její podrobnější popis a příklady jsou pro tuto práci nadbytečné.

Pro kontext této práce je vhodné tento pojem chápat v širším kontextu, tedy **coby odstraňování redundance i již existujících dat** (ne pouze schématu, které ani nemusí vždy existovat) a to tedy i mimo RDBS.

Možným omezením normalizace je její přímá možná aplikace výhradně na strukturovaná data, u kterých je případná redundance jednoznačně klasifikovatelná, ačkoli za normalizaci lze označit i lidské úpravy nestrukturovaného textu. Z tohoto důvodu může proces normalizace dat přímo následovat proces strukturalizace, což v praxi odpovídá například procesu analýzy a návrhu systémů, u nichž je zpravidla vycházeno z nestrukturovaného zadání. Spolu s normalizací dále může docházet k dehierarchizaci, jak již uvedla předchozí sekce 7.2.4.

Prakticky lze za normalizační proces pokládat každý, při němž stoupá míra (hustota) informace datového objektu, tak jak byla definována v sekci 5.4 a strukturovanost tedy není nutnou podmínkou.

Normalizace je **slučitelná se strukturalizačním procesem**, ačkoli destrukturalizaci nemusí nutně vylučovat. Na míru hierarchičnosti může mít výhradně **negativní vliv** a je tedy slučitelná s dehierarchizací, zatímco **hierarchizaci vylučuje**.

7.2.6 Denormalizace

Obecně je denormalizačním procesem takový proces, při němž míra informace datového objektu klesá.

Denormalizace, tedy proces vytváření redundance, může mít řadu příčin, potažmo zdůvodnění, v závislosti na své úmyslnosti. K **úmyslné denormalizaci** obvykle dochází v rámci procesu, kdy jsou data připravována k použití, například k prezentaci pro člověka (stejný případ jako v předchozí sekci 7.2.2). V tomto případě je cílem vyšší zpracovatelnost lidským mozkem na úkor stroje a lze předpokládat možnou paralelní hierarchizaci; možná je i destrukturalizace. Úmyslné redundance byly dále již popsány v sekci 5.4. K **neúmyslné denormalizaci** může dojít v rámci z nějakého důvodu špatně navrženého nebo nedostatečně popsaného transformačního procesu. V tomto případě dotyčný pokles míry informace plní funkci upozornění na tento jev.

Denormalizace má často **negativní vliv na míru strukturovanosti** (ovšem nevylučuje teoreticky ani strukturalizaci) a pravděpodobný **pozitivní vliv na míru hierarchičnosti** (vylučuje dehierarchizaci – jsou-li některá data duplikována, nemůže tím vzniknout nehierarchická vazba).

7.3 Kombinované transformace

Tato část uvádí řadu kombinací změn jednotlivých metrik a diskutuje jejich slučitelnost na základě vlastního rozboru jejich podstaty.

Jak již nastínily předchozí sekce týkající se šesti základních druhů datových transformací (značeny S, DS, H, DH, N, DN, jak uvádí již Tabulka 3), mají spolu tyto transformace často úzký vztah. Vzájemně se ovlivňují a mohou společně korelovat, či naopak se vylučovat. V praxi velice často dochází k jejich kombinaci, kdy se mění hodnoty více než jedné metriky, zatímco metriky ostatní se nemění vůbec, popřípadě jejich změna není známa.

Matematicky je možné datovou transformaci datového objektu D ze stavu D_0 do stavu D_1 vyjádřit tzv. **transformačním vektorem**, zachycujícím změny hodnot metrik v průběhu transformace, jak ukazuje Vzorec 12.

$$\underset{t}{\rightarrow}(D) = \left({}_{\Delta}s(D), {}_{\Delta}h(D), {}_{\Delta}i(D) \right) = \left(s(D_1) - s(D_0), h(D_1) - h(D_0), i(D_1) - i(D_0) \right)$$

Vzorec 12 – datový transformační vektor

Na základě tohoto vektoru lze spočítat tzv. **délku transformace**, vyjadřující obecnou míru rozdílnosti (nekompatibility) obou stavů, což znázorňuje Vzorec 13. Hodnota je opět škálována na interval $\langle 0; 1 \rangle$ a je tedy dělena maximální možnou **geometrickou délkou** úsečky v krychli trojrozměrného metadatového prostoru, kterou je tělesová úhlopříčka o délce $\sqrt{3}$.

$$t(D_0, D_1) = \frac{1}{\sqrt{3}} \cdot \sqrt{\left(s(D_1) - s(D_0)\right)^2 + \left(h(D_1) - h(D_0)\right)^2 + \left(i(D_1) - i(D_0)\right)^2}$$

Vzorec 13 – délka datové transformace

Tato hodnota má vztah k **míře kompatibility** (část 7.1), na rozdíl ní ovšem nevyjadřuje vztah k obecnému typu datové struktury, ale k reálnému výslednému datovému objektu. Délku výsledné transformace $t(D_0, D_1)$ lze porovnat s hodnotou $1 - k(D, S)$. Pokud transformace proběhla očekávaným způsobem a nebylo nutné data modifikovat jinak než na účelem jejich kompatibility s cílovým formátem, měly by obě hodnoty být podobné.

Pokud prozatím neuvážíme vztahy mezi základními transformacemi uvedenými v části 7.2, lze určit výchozí **počet kombinovaných transformací**, a to na základě počtu možných kombinací základních šesti druhů. Jsou-li vyloučeny kombinace obsahující **protichůdné transformace** (například strukturalizace a destrukturalizace, které naráz pochopitelně nedávají smysl), může se jednat o jednice, dvojice nebo trojice. Lze si představit i existenci transformace, kdy se nemění žádná z metrik (resp. její změna není známa, jedná se tedy o nulový vektor) – tu nelze vyloučit, ovšem pro tuto práci **nemá žádný význam a není dále uvažována ani započítána**.

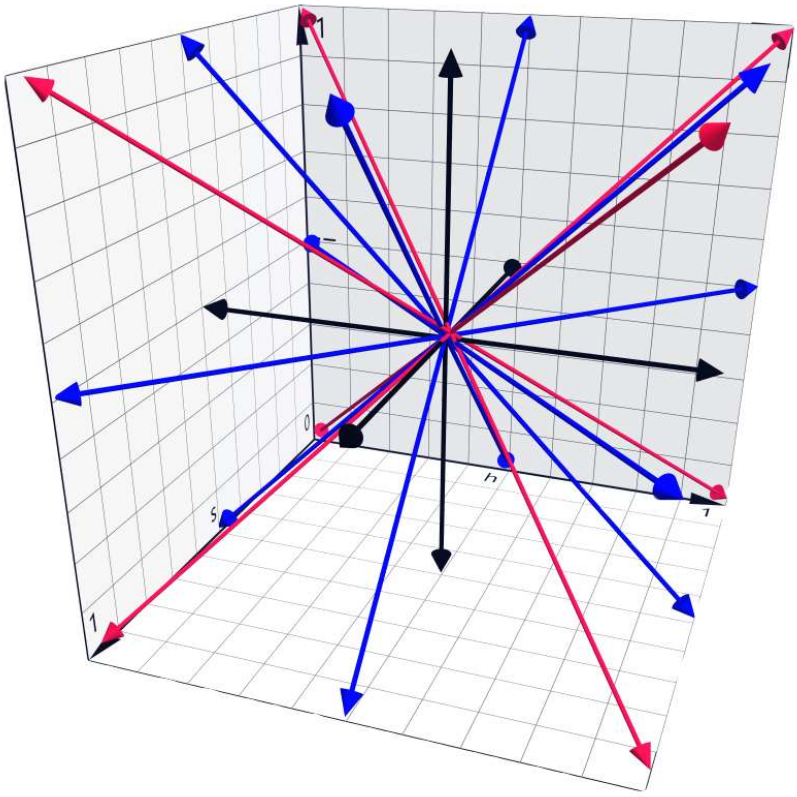
Počet všech kombinací lze spočítat jako součet všech těchto n-tic – v následujícím vzorci ve vzestupném pořadí, kdy od počtu kombinací je odečten počet kombinací obsahujících protichůdné transformace.

$$\binom{6}{1} + \left(\binom{6}{2} - 3\right) + \left(\binom{6}{3} - 12\right) = 26$$

Vzorec 14 – výchozí počet možných směrů transformací

Tento výpočet tedy odpovídá počtu možných hlavních směrů pohybu v trojrozměrném, tedy i v této práci používaném trojrozměrném metadatovém prostoru – pokles a vzestup některé ze tří metrik a kombinace těchto poklesů a vzestupů. Tyto směry znázorňuje Ob-

rázek 19. Černé šipky mířící ke **stěnám** pomyslné krychle znázorňují změnu **jedné metriky** (základních šest typů transformací). Modré šipky mířící k **hranám** znázorňují změnu **dvou metrik** (dvanáct kombinovaných dvojitých transformací) Červené šipky mířící k **rohům** znázorňují změnu **tří metrik** (osm kombinovaných trojitých transformací).



Obrázek 19 – hlavní možné směry datové transformace znázorněné v trojrozměrném metadatovém prostoru

Těchto možných 26 směrů ovšem nebere v potaz **faktické vlivy** změn některých metrik na metriky ostatní, jak popsala diskuse v závěrečných odstavcích sekcí 7.2.1 až 7.2.6. V praxi některé změny na osách více metrik jsou navzájem do jisté míry úměrné (přímo či nepřímo) a tedy **slučitelné**, popřípadě mají neutrální vztah nebo se vylučují. Vzhledem k existenci vyloučených kombinací je zřejmé, že reálných možných kombinovaných transformací je nižší počet než kombinatoricky vypočítaných 26.

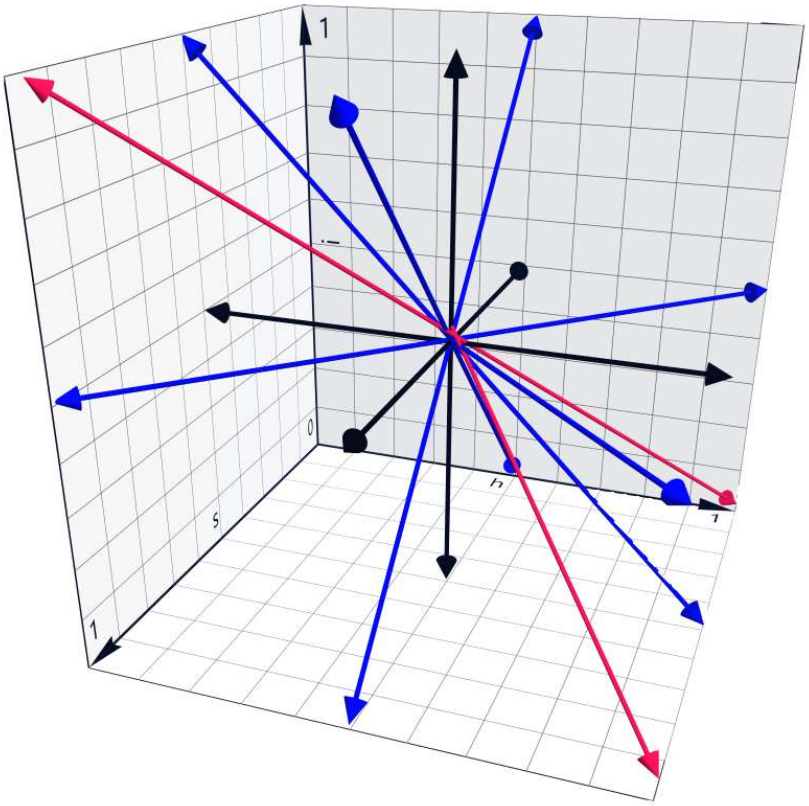
Kombinace	**Slučitelnost**	**Vysvětlení**
S, H	Ano	Strukturalizace umožňuje veškeré ostatní typy transformací, viz sekce 7.2.1
S, N	Ano	

S, DH	Ano	
S, DN	Ano	
H, N	Ne	Hierarchizace ruší některé vazby a vynucuje duplikování dat, viz sekce 7.2.3 a článek (Vodňanský 2016a)
H, DN	Ano	
H, DS	Ano	Při hierarchizaci jde míru strukturovanosti zvyšovat i snižovat, viz sekce 7.2.2 a 7.2.3
N, DS	Ano	Nelze teoreticky vyloučit normalizaci dat při jejich převodu do nestrukturované podoby, viz sekce 7.2.2 a 7.2.5
N, DH	Ano	Při normalizaci mohou být nevhodně hierarchická data dehierarchizována tvorbou vazeb, viz sekce 7.2.4 a 7.2.5
DS, DN	Ano	Typický paralelní proces, viz sekce 7.2.2 a 7.2.6
DS, DH	Ne	Méně strukturovaná data mohou méně porušit definici hierarchičnosti, viz sekce 7.2.2 a 7.2.4
DN, DH	Ne	Tvorba nehierarchických vazeb nemůže probíhat v rámci duplikování některých dat při denormalizaci, viz sekce 7.2.4 a 7.2.6
S, H, N	Ne	Je vyloučena kombinace **H, N**
S, H, DN	Ano	Odpovídající dvojice jsou slučitelné
S, DH, N	Ano	Odpovídající dvojice jsou slučitelné
DS, H, N	Ne	Je vyloučena kombinace **H, N**
S, DH, DN	Ne	Je vyloučena kombinace **DN, DH**
DS, H, DN	Ano	Odpovídající dvojice jsou slučitelné
DS, DH, N	Ne	Je vyloučena kombinace **DN, DH**
DS, DH, DN	Ne	Je vyloučena kombinace **DN, DH**

Tabulka 4 – souhrn slučitelnosti kombinovaných datových transformací dle sekcí 7.2.1 až 7.2.6

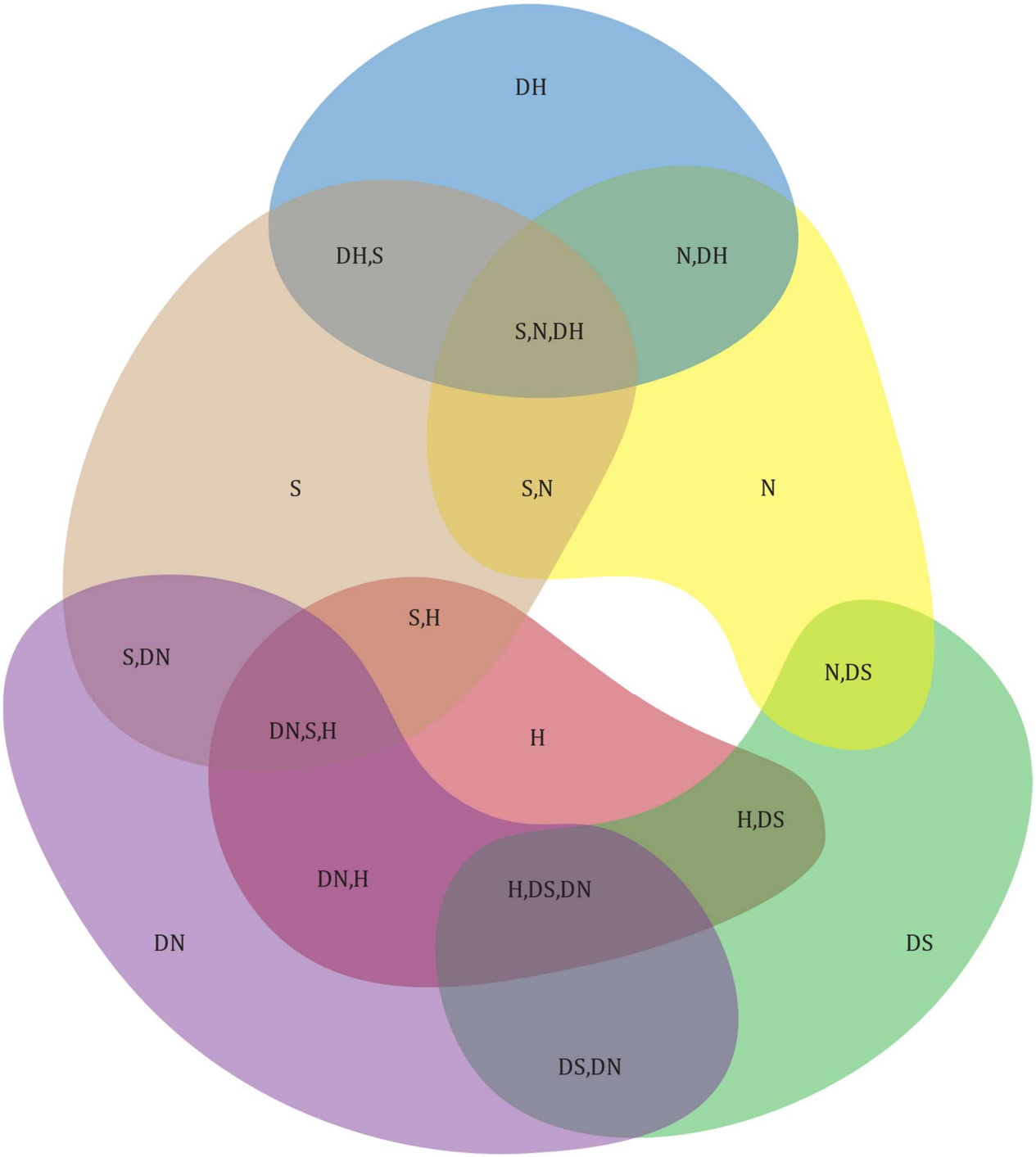
Dle závěrů sekcí 7.2.1 až 7.2.6, které shrnuje Tabulka 4, je reálně možných kombinací pouze 18, pokud odstraníme 8 neslučitelných kombinací (poznamenejme, že 6 základních, nekombinovaných transformací tabulka neuvádí).

Obrázek 19 je tedy možné zredukovat – odstraněním těchto neslučitelných kombinací vzniká Obrázek 20.



Obrázek 20 – prakticky slučitelné směry datové transformace v trojrozměrném metadatovém prostoru

Takovýto počet je již dostatečně nízký na to, aby bylo možné ho poměrně přehledně znázornit Vennovým diagramem, jak ukazuje Obrázek 21 – zkratky oddělené čárkou značí transformace stejným způsobem jako Tabulka 3 a Tabulka 4.



Obrázek 21 – Vennův diagram kombinovaných datových transformací

Vizualizace pomocí Vennova diagramu umožňuje poměrně přehledným způsobem umístit datovou transformaci do diagramu dle jejího typu, popřípadě transformace navzájem porovnávat. V tomto diagramu nelze znázornit, do jaké míry k dotyčnému posunu metrik dochází. Toto může být potenciálně do budoucna řešitelné pomocí Vennova diagramu fuzzy množin, jaký navrhuje např. (Loh a Subramanian 2010). Vennův diagram je nicméně vhodný zejména pro klasifikaci zcela jednoznačně zařaditelných transformací.

8 Modelování metadatového skladu

Tato kapitola popisuje obecné schéma metadatového skladu použitelné pro výpočet a uložení hodnot metrik pro různé datové objekty. Zabývá se především vymezením hlavních odlišností vůči principu OLAP, specifikuje datový model skladu a uvádí základní principy jeho budování a plnění naměřenými hodnotami.

Název „metadatový sklad“ se vymezuje vůči tradičnímu pojmu „datový sklad“ (případně příbuzným „datovým tržištím“) přidáním předpony „meta-“. Nejedná se tedy o přeuspořádaná původní transakční data, nýbrž o obdobným způsobem uložená „data o datech“, v tomto případě představujícím vypočtené hodnoty metrik a s nimi související údaje.

8.1 Vztah k OLAP a hlavní odlišnosti

Základní princip **vychází z konceptu OLAP**, tedy analytické databáze skládající se z centrálních tabulek faktů, jejichž řádky odkazují na dimenze v dotyčných tabulkách. Tento typ struktury je také znám jako „star“, popřípadě „snowflake“ a to na základě tvaru schématu.

Koncept metadatového skladu se ovšem do značné míry odlišuje. **První odlišností** je větší počet naráz zkoumaných hodnot (konkrétně dle této práce dotyčné tři), plnících roli jak metrik, tak také dimenzí, jak již zdůvodnila kapitola 3. Z hlediska OLAP se tedy jedná o tzv. degenerovanou dimenzi – hodnotu obsaženou již přímo v tabulce faktů bez vazby na tabulku dimenzí (Becker 2003).

Druhou odlišností je rozdílný způsob agregace hodnot. Zatímco v běžných OLAP lze hodnoty napříč dimenzemi agregovat přímo prostřednictvím matematických funkcí, jako je průměr, minimum nebo maximum, v případě v této práci zmíněných metrik může být situace zcela odlišná. Například může existovat několik paralelně datových jednotek vnitřně zcela neredundantních, ovšem navzájem identických či velmi podobných a utvářejících tak silnou vnější redundanci. Při jejich samostatném zkoumání je tak míra informace diametrálně odlišná od zkoumání celého datového objektu, v němž jsou řazeny. Stejný problém platí i u hierarchičnosti (porušení hierarchičnosti vzniká při jistých kombinacích objektů) a strukturovanosti (organizováním objektů do nadobjektů míra strukturovanosti narůstá). Namísto agregace jsou tak v každém datovém objektu hodnoty počítány zvlášť, což může být výpočetně náročné.

V důsledku této výpočetní náročnosti problému se nabízí otázka, zda je nutné, aby veškeré hodnoty byly spočítány předem v metadatovém skladě pro celou hierarchii datových objektů. Nejvhodnějším přístupem při vyšších objemech dat může být výpočet hodnot pouze pro elementární datové objekty a na vyšších úrovních výpočet ad-hoc v okamžik dotazu, popřípadě ryzí ad-hoc výpočet. Ad-hoc přístup má význam zejména v situaci, je-li zkoumán pouze izolovaný stav či jev v datové základně a nikoli základna jako celek.

Třetí odlišností je, že klasické datové sklady (jejich schéma) jsou zpravidla budovány zvlášť pro každé konkrétní praktické užití – jednotlivé organizace zpravidla zkoumají odlišné a zejména odlišně dimenzionálně členěné hodnoty. Oproti tomu metadatový sklad má většinu konceptů i základní datové schéma obecně použitelné (ačkoli dodatečné změny nikdy nelze vyloučit). Lze tedy předpokládat jeho možné užití na různých datech. Princip skládání datových objektů, jejich napojení na samotná data a transformační proces (obdobu datové pumpy a ETL procesů známých z OLAP) je ovšem z těchto důvodů nutné navrhnout ad-hoc a je proto nutný vstup **experta**. Jeho součástí je i rozhodnutí, nakolik a jakým způsobem budou tyto procesy automatizovány. V případě menších objemů (popřípadě nižší komplexity) dat a obtížnějšího stanovování mechanismu výpočtu hodnot nelze vyloučit ani zcela manuální tvorbu skladu.

8.2 Základní schéma

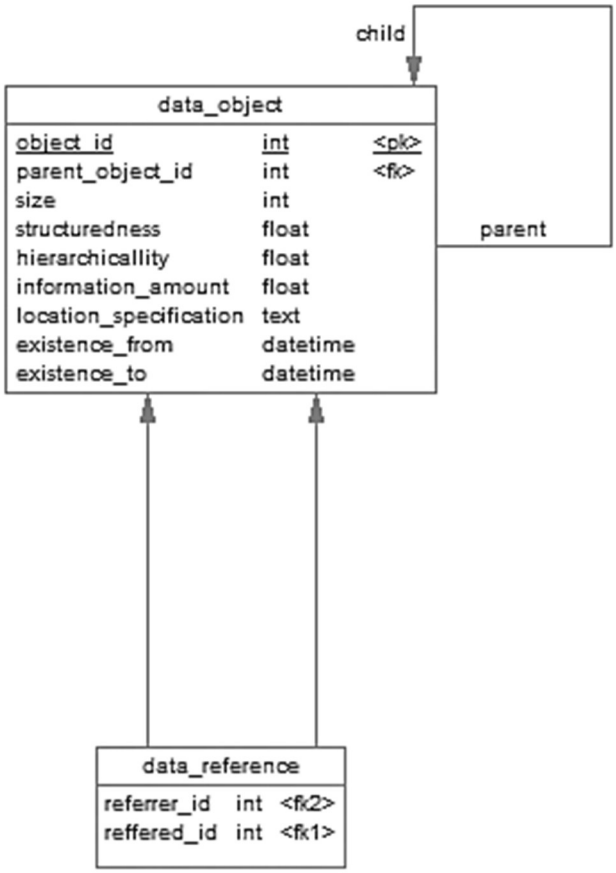
Jedním ze základních pravidel metadatového skladu je **reference datových objektů na dotyčná reprezentovaná data**, pokud v dotyčném čase existují (v opačném případě reference na jejich umístění v čase existence) – sloupec `location_specification`. Protože odkazovaná data (jejich typ datové struktury) jsou pochopitelně heterogenní, nelze formát referenčního mechanismu předem specifikovat pro všechny možné případy. Je ovšem možné ho nastínit pro nejtypičtější datové struktury – u relačních databází se tedy jedná o identifikátor databáze, potažmo tabulky, řádku v ní až konkrétní hodnoty. V případě formátu XML jde o odkaz na dotyčné elementy v konkrétních XML dokumentech. U nestrukturovaných dat může jít o prosté odkazy na soubory. V případě RDF pak o identifikátory URI. Referujícím subjektem je vždy každý datový objekt a data pomocí něj musí být možné jednoznačně dohledat v čase jejich existence.

Je také nutné podotknout, že zkoumaný datový objekt nemusí reprezentovat fyzicky uložená data a jeho úložiště tak nemusí vyjadřovat umístění. Namísto něj v těchto případech

zpravidla jde o výsledek určitého procesu generujícího příslušná data a právě tento proces je místo toho identifikován. Typickým příkladem v tomto případě je URL adresa endpointu na serveru nebo i přímo funkce v kódu aplikace.

Vzhledem k univerzálnímu konceptu metadatových skladů je proto nutné referenční mechanismus specifikovat zvlášť pro každou datovou základnu, neboť různých typů datových struktur, a tedy i způsobů odkazování na ně je teoreticky neomezený počet.

Výsledný metadatový sklad je reprezentován relační databází, a to o následující struktuře (Obrázek 22).



Obrázek 22 – fyzický model metadatového skladu

Metadatový sklad je relační databází o dvou tabulkách, z nichž jedna je vazební. Datový objekt je v rekurzivní relaci, kdy sloupec `parent_object_id` je cizím klíčem odkazují-

cím na `object_id` nadřazeného objektu. Pro ten je určována prostá bitová velikost, hodnoty metrik a specifikace fyzického umístění – unikátní odkaz na reprezentovaná data (popsaný již v prvním odstavci této sekce). Formát tohoto odkazu je závislý na konkrétní datové struktuře a může být

- **relativní** (vymezující se s ohledem na specifikaci umístění rodičovského objektu) nebo
- **absolutní** (celkové umístění v datové struktuře) – ten je v práci dále používán pro lepší přehlednost.

Dále je určována časová platnost existence dat odpovídajících objektu. Ta má význam zejména při zkoumání datové transformace nebo v případě dlouhodobějšího monitorování dat; při statickém jednorázovém výpočtu mohou být tyto hodnoty ignorovány nebo nahrazeny pouze jednou hodnotou odpovídající momentu výpočtu.

Dále datový objekt obsahuje 2 typy rekurzivních vazeb.

První typ vazby je odkaz na rodičovský objekt o kardinalitě 1: N (rodič : dítě). Tato vazba specifikuje náležitost objektu do hierarchie fyzického úložiště zkoumané datové základny; například řádek tabulky RDBS je dítětem této tabulky, XML element je dítětem nadřazeného elementu nebo prostý soubor je dítětem své složky. Zde existuje pouze jediný objekt neodkazující na žádného rodiče a tím je hlavní datový objekt specifikující zkoumanou datovou základnu jako celek. Naopak nikým neodkazované objekty (listy pomyslného stromového grafu) jsou dále nedělitelné, tedy elementární objekty. Každý objekt pak má vypočteny hodnoty metrik pro celou svou subhierarchii. Na této vazbě nesmí vzniknout mezi daty cyklus.

Druhý typ vazby je poté vzájemné explicitní odkazování samotných dat. Zde může libovolný objekt odkazovat i být odkazován s ostatními a o hierarchii se tak v žádném případě jednat nemusí. Tato vazba zachycuje například odkazy položek cizích klíčů v tabulkách RDBS, URI identifikátory v RDF datech, reference uvnitř XML, popřípadě HTML a libovolné jiné odkazy napříč datovou základnou (s umožněním cyklů). Tato vazba slouží zejména k výpočtu hierarchičnosti datových objektů; umožňuje zakreslení datových objektů do jednoduchého orientovaného (v tomto případě ne nutně stromového) grafu. U tohoto typu vazby je velmi důležité rozlišit **formalitu**:

- **formální vazba** je vazbou vymezenou v oficiálních specifikacích a standardech dotyčného datového formátu a může se jednat o
    - cizí klíče v relační databázi,
    - reference specifikované schématy XML, např. keyref, které ovšem lze pokládat za částečně nekoncepční z důvodů uvedených v části 6.2,
    - URI identifikátory v RDF,
- **neformální vazba** je odkazování dat způsobem, který dotyčný datový formát nepodporuje, popřípadě jím podporované mechanismy nejsou využity. Například jde o odkaz jedné tabulky na primární klíč jiné tabulky v RDBS bez specifikace cizího klíče. Jejich odhalení a zpracování do datového skladu může být obtížné i nemožné a lze předpokládat, že ne vždy jsou nalezeny nebo uznány jako součást datové struktury.

U vzájemného odkazování dat je nutné zohlednit, který datový objekt reprezentuje odkazující entitu – u něj musí být tato vazba v metadatovém skladu vytvořena. Například cizí klíč v řádku RDBS je odkazem dotyčného řádku, nikoli sloupce obsahujícího cizí klíč. Podstatné je, aby u datového objektu bylo možné detekovat porušení definice hierarchičnosti.

Dále je podstatné, že druhý typ vazby se může překrývat s prvním, tedy že data jsou zároveň uložena v hierarchii rodič – dítě a jedná se i o informační vazbu o dotyčné kardinalitě. Oba typy vazeb ovšem nesmí být zaměňovány.

Časový atribut, tedy platnost datového objektu vymezená dvěma časy, specifikuje okamžiky, v jejichž průběhu byl objekt bez jakékoli změny jednorázově (pokud se okamžiky shodují) nebo vícenásobně načten do metadatového skladu. Při jakékoli změně dat odpovídajících objektu (tedy i kterémukoli jeho podobjektu), popřípadě jejich zániku musí být vytvořen datový objekt nový. Časový atribut nemusí být při jednorázovém zkoumání dat přítomen, popřípadě ho lze nahradit periodickým kompletním znovunačítáním, vymezeným vždy jedním dotyčným datem a časem, což je výrazně jednodušší vzhledem k tomu, že není nutné porovnávat změny dat, ovšem také náročnější na velikost celého skladu.

8.3 Obecný postup budování

Postup tvorby metadatového skladu lze obecně popsat následujícími šesti kroky. Tyto kroky následně se zohledněním specifik konkrétních dat aplikuje část 10.2 a 10.3.

1. **Tvorba stromu datových objektů**. Do databáze metadatového skladu jsou vkládány samotné objekty se svou identifikací umístění, a to od hlavního datnového objektu k nižším, až elementárním podobjektům.

2. Plnění **vazební tabulky** `data_reference`. Je definováno, který jev bude pokládán za datovou referenci a umožňuje tak porušení definice hierarchičnosti.

3. Výpočet a vložení **velikosti** datových objektů. Zpravidla půjde o velikost binárních dat nebo o textovou délku, pokud to formát veškerých dat umožňuje svým textovým charakterem.

4. Výpočet a vložení hodnot **míry hierarchičnosti** na základě vazební tabulky z druhého kroku. Pro data, jejichž formát neumožňuje porušení definice hierarchičnosti je automaticky vložena hodnota 1.

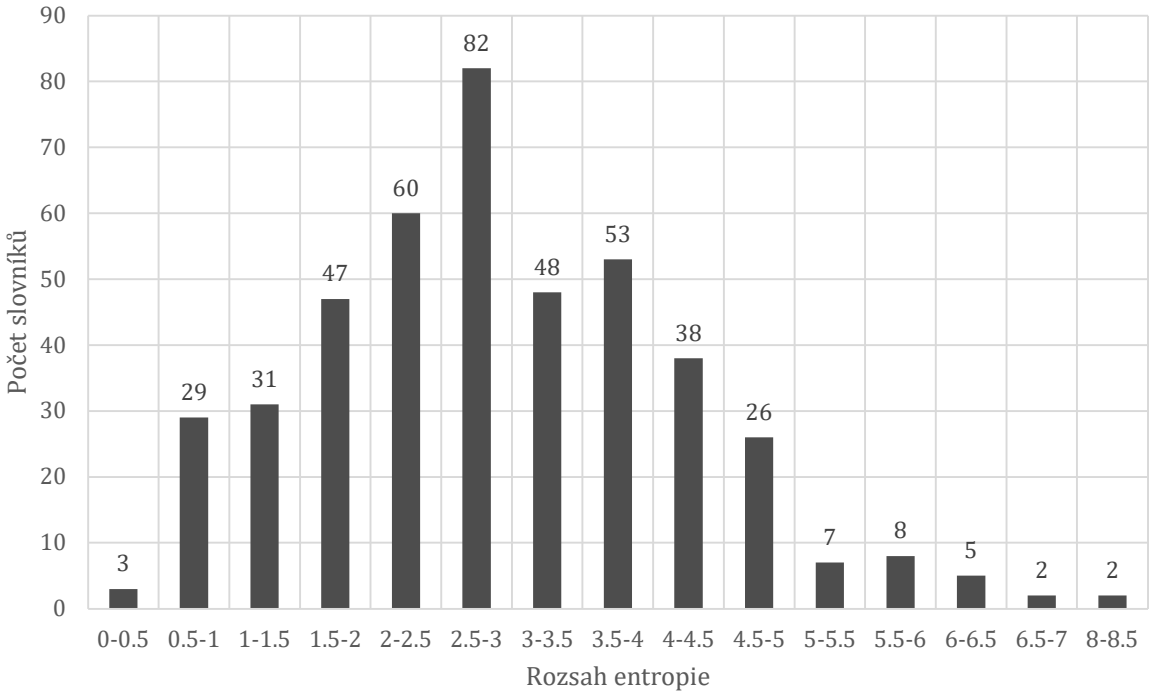
5. Výpočet a vložení hodnot **míry strukturovanosti**. Zde musí být postupováno od elementárních datových objektů, u nichž lze tuto hodnotu stanovit jednoznačným kritériem dále k vyšším úrovním datových objektů.

6. Výpočet a vložení hodnot **míry informace.** Výpočetně jde zřejmě obvykle o nejnáročnější krok vzhledem k nutnosti komprimovat často velmi velká data a je proto prováděn v závěru.

9 Ukázková analýza informačních struktur

Hlavním vybranou ukázkou informačních struktur jsou kolekce ontologických slovníků OntoFarm (Svátek et al. nedatováno) a LOV – Linked Open Vocabularies (Vandenbussche et al. 2017). Zde předpokládáme, že ontologický slovník náleží informační úrovni, jak bylo diskutováno již v části 4.5.

Tato kapitola popisuje výpočet dvou metrik na základě kolekcí ontologických slovníků a jedná se tedy o T-Box – slovníky vymezují terminologii a neuvádí konkrétní fakta. Výpočet míry informace je proto do jisté míry problematický, jak již popisuje sekce 4.2.3. Dalším již zmíněným problémem je, že při výpočtech nelze pracovat s informací samotnou, nýbrž výhradně s její určitou datovou reprezentací. Ontologie a ontologické slovníky lze ovšem chápat jako velmi přesný způsob reprezentování informace či jejího schématu ve srovnání s jinými typy datových struktur, které tato práce uvádí.



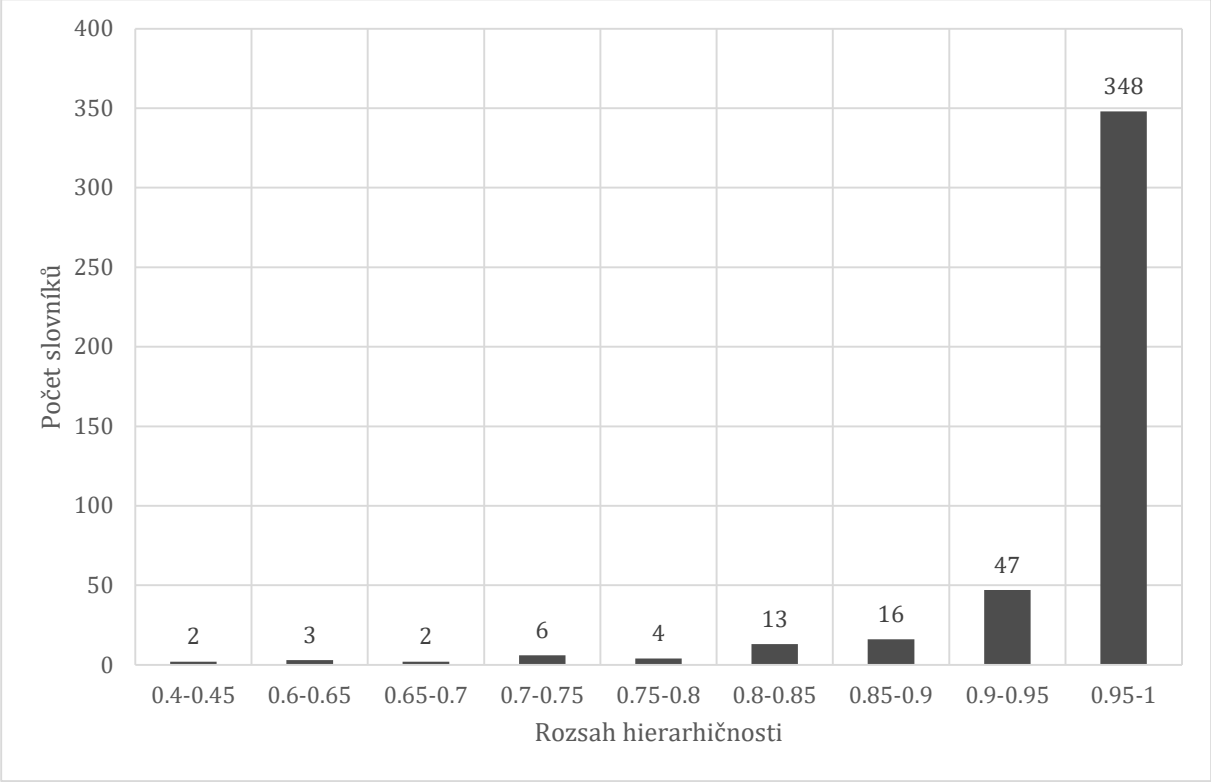
Obrázek 23 – histogram entropie ontologických slovníků

V publikaci (Vodňanský a Zamazal 2016) jsme se zabývali výpočtem entropie dle (Doran et al. 2008) v rámci kolekce ontologických slovníků uvedených na začátku této kapitoly. K výpočtu byl použit Vzorec 3, tedy stupeň vrcholu grafové reprezentace ontologického

slovníku (orientovaný graf tříd a jejich relací typu *subClassOf*), což v tomto případě reprezentuje míru informace navzdory jisté odchylce od způsobu, jakým je pojata v této práci (aplikující AIT). Výslednou průměrnou hodnotou je 3,970 pro kolekci OntoFarm a 2,876 pro kolekci LOV. Zajímavějšího znázornění výsledku je ovšem dosaženo, pokud pro intervaly hodnot entropie o velikosti 0,05 jsou zkoumány počty odpovídajících slovníků. Výsledný histogram hodnot obou kolekcí ukazuje Obrázek 23.

Je patrné, že rozložení je blízké normálnímu (gaussovskému) s převahou středních hodnot a malým podílem hodnot okrajových, kdy nejvíce ontologických slovníků má entropii v rozsahu 2,5 až 3, přičemž průměrná hodnota entropie za obě uvedené kolekce je 3,19.

Druhou měřenou hodnotou v (Vodňanský a Zamazal 2016) je míra hierarchičnosti. Dle předpokladu dosahuje zpravidla hodnot velmi se blížících jedné - 98.9% pro kolekci OntoFarm a 96.6% pro kolekci LOV. Kritériem je Vzorec 1 na str. 29 v sekci 4.2.2 a definice tak zcela odpovídá; je ovšem nutné podotknout, že míra hierarchičnosti dat odpovídajících ontologickému slovníku se může i velmi výrazně lišit. I hodnoty míry hierarchičnosti lze zanést do histogramu, jak znázorňuje Obrázek 24.



Obrázek 24 – histogram míry hierarchičnosti ontologických slovníků

V naprosté většině případů (79%) tedy je míra hierarchičnosti vyšší než 0,95, ovšem pouze v 70% měřených slovníků je ovšem míra hierarchičnosti absolutní, tedy rovna 1.

Naopak jen velmi zřídka je míra hierarchičnosti nižší než 0,7. Nízké hodnoty navzdory svému výrazně menšinovému podílu jsou jevem, který se opakuje a dokáže klasifikovat jistý drobný podíl ontologických slovníků, jejichž míra hierarchičnosti je nezvykle nízká, což má vliv na některé procesy datových transformací, které zkoumá kapitola 7. Je patrné, že histogram hodnot se blíží Poissonovu rozdělení s parametrem $\lambda = 1$.

Míra strukturovanosti v této části práce počítána nebyla. Její pojetí coby rozložitelnost datových objektů je v rozporu se základním konceptem informačních ontologií a jejich slovníků, které jsou od podstaty určené ke zcela strukturovanému ukládání (lze je pokládat i za významný nástroj strukturalizace dat). Její výpočet tak u těchto kolekcí nedává smysl. Do budoucna ovšem nelze vyloučit výzkum problematiky zaměřené na vliv různých typů ontologických slovníků na odlišně pojatou míru strukturovanosti jim odpovídajících dat.

10 Ukázková analýza datových struktur

10.1 Výběr vzorku dat a základní popis

Princip výpočtu hodnot metrik a jejich organizace do metadatového skladu je ukázán na dvou typech dat. Demonstrační data jsou vytvořena pro znázornění principu výpočtu hodnot metrik. Následně reálná data ukazují automatizaci celého výpočtu na rozsáhlejším vzorku.

10.1.1 Demonstrační data

Demonstrační data (značená dále DD) jsou účelově vybraná či přímo vytvořená velmi malá data, jejichž účelem je zejména srozumitelné představení základních principů co nejjednodušším možným způsobem. Přitom bylo dbáno na co nejmenší rozsah, aby celý výpočet mohl být kompletně proveden bez nutnosti automatizace. Dalším hlediskem byla vzájemná odlišnost dat, co se týče předpokládaných hodnot metrik.

První demonstrační data DD1 jsou tvořena relační databází vycházející ze schématu, které znázornil již Obrázek 17 – velice jednoduchý model o 3 tabulkách propojených klíči. Jejich obsah je následující:

id_area	name	description
1	swimming pool	This is an example long description of a swimming pool
2	football pit	
3	tenis court	Tenis

Tabulka 5 – tabulka „Area“ RDBS prvních demonstračních dat DD1

id_course	day	time	id_area	id_period
1	1	15:00:00	1	3
2	2	15:00:00	3	3
3	5	14:40:00	3	3
4	2	14:00:00	1	4
5	3	15:30:00	2	4

Tabulka 6 – tabulka „Couse“ RDBS prvních demonstračních dat DD1

id_period	from	to
3	2019-03-01	2019-06-30
4	2018-09-01	2019-02-26

Tabulka 7 – tabulka „Period“ RDBS prvních demonstračních dat DD1

Druhá demonstrační data DD2 jsou tvořena fragmentem HTML stránky. Stránka není pro jednoduchost zkoumána celá – některé části obsahující metadata nebo skripty jsou komplikované a nemají sémantický význam, který by pomohl názornosti tohoto příkladu. Jedná se o reálná data z webu Fakulty informatiky a statistiky VŠE. Tento fragment webové stránky byl vybrán především pro zjevnou heterogenitu míry strukturovanosti jednotlivých datových podobjektů (HTML elementy), která je zastoupena ve všech třech hlavních stupních (strukturovaná, semistrukturovaná a nestrukturovaná data).

```
<div class="table-responsive">
    <table class="table table-striped" style="width: 800px;">
        <thead></thead>
        <tbody>
            <tr>
                <td style="text-align: left; width:
10%"><strong>pondělí</strong></td>
                <td style="text-align: left; width: 15%">8:30 – 11:30</td>
                <td style="text-align: centr; width: 5%">a</td>
                <td style="text-align: left; width: 20%">13:00 – 15:00</td>
                <td style="text-align: left; width: 45%">všechny
referentky</td>
            </tr>
            <tr>
                <td style="text-align: left;"><strong>středa</strong></td>
                <td style="text-align: left;">8:30 – 11:30</td>
                <td style="text-align: centr;">a</td>
                <td style="text-align: left;">13:00 – 15:00</td>
```

```
                <td style="text-align: left;">Ing. J. Sedláčková, I.
Hudcová, J. Krajíčková</td>
            </tr>
            <tr>
                <td style="text-align: left;"></td>
                <td style="text-align: left;">8:30 – 11:30</td>
                <td style="text-align: centr;">a</td>
                <td style="text-align: left;">12:30 – 14:30</td>
                <td style="text-align: left;">J. Hudčeková</td>
            </tr>

        </tbody>
    </table>
    <hr>
    <p style="text-align: justify;"><strong>Upozornění</strong>! V úterý a
ve čtvrtek úřední hodiny nejsou! V těchto dnech je možné vyřizovat
záležitosti <strong>pouze na základě předchozí domluvy se svou studijní
referentkou</strong>. Prázdninové úřední hodiny naleznete vždy koncem
června v Aktualitách FIS.</p>
    <hr>
    <table style="width: 716px; height: 101px;" border="0" cellspacing="0"
cellpadding="0" class="table">
        <thead>
            <tr valign="top">
                <th bgcolor="#efefef" width="227"><strong>Ing. Jana
Sedláčková</strong>
                    <br> (bakalářské studium)
                    <p></p>
                    <ul type="none">
                        <li>místnost: 411 NB</li>
                        <li>telefon: 224 095 441</li>
                        <li>e-mail: <a
href="mailto:sedlacko@vse.cz">sedlacko@vse.cz</a></li>
                    </ul>
                </th>
                <th bgcolor="#efefef" width="227"><strong>Jana
Hudčeková</strong>
                    <br> (bakalářské studium)
                    <p></p>
                    <ul type="none">
                        <li>místnost: 410 NB</li>
                        <li>telefon: 224 095 440</li>
                        <li>e-mail: <a
href="mailto:jana.hudcekova@vse.cz">jana.hudcekova@vse.cz</a></li>
                    </ul>
                </th>
            </tr>
        </thead>
        <tbody>

        </tbody>
    </table>
    <p> </p>
</div>
```

Kód 1 – fragment HTML stránky coby druhá demonstrační data DD2 (Vysoká škola ekonomická v Praze, nedatováno)

Třetí demonstrační data DD3 jsou založena na formátu JSON. Jedná se o fiktivní data do značné míry se shodující s prvními demonstračními daty DD1, ovšem uspořádaná záměrně tím způsobem, aby hierarchicky odpovídala formátu JSON.

```
{
   "coursesByArea":{
      "swimming_pool":{
         "description":"This is an example long description of a swimming
pool",
         "courses":[
            {
               "day":1,
               "time":"15:00",
               "period":"2019-03-01 - 2019-06-30"
            },
            {
               "day":2,
               "time":"14:00",
               "period":"2018-09-01 - 2019-02-26"
            }
         ]
      },
      "football_pit":{
         "description":"",
         "courses":[
            {
               "day":3,
               "time":"15:30",
               "period":"2018-09-01 - 2019-02-26"
            }
         ]
      },
      "tenis_court":{
         "description":"Tenis",
         "courses":[
            {
               "day":2,
               "time":"15:00",
               "period":"2019-03-01 - 2019-06-30"
            },
            {
               "day":5,
               "time":"14:40",
               "period":"2019-03-01 - 2019-06-30"
            }
         ]
      }
   },
   "coursesByPeriod":[
      {
         "range":"2018-09-01 - 2019-02-26",
         "courses":[
            {
               "day":1,
               "time":"15:00",
               "area":"swimming_pool"
            },
            {
               "day":2,
```

```
                "time":"15:00",
                "area":"tenis_court"
            },
            {
                "day":5,
                "time":"14:40",
                "area":"tenis_court"
            }
        ]
    },
    {
        "range":"2019-03-01 - 2019-06-30",
        "courses":[
            {
                "day":2,
                "time":"14:00",
                "area":"swimming_pool"
            },
            {
                "day":3,
                "time":"15:30",
                "area":"football_pit"
            }
        ]
    }
  ]
}
```


Kód 2 – objekt JSON coby třetí demonstrační data DD3

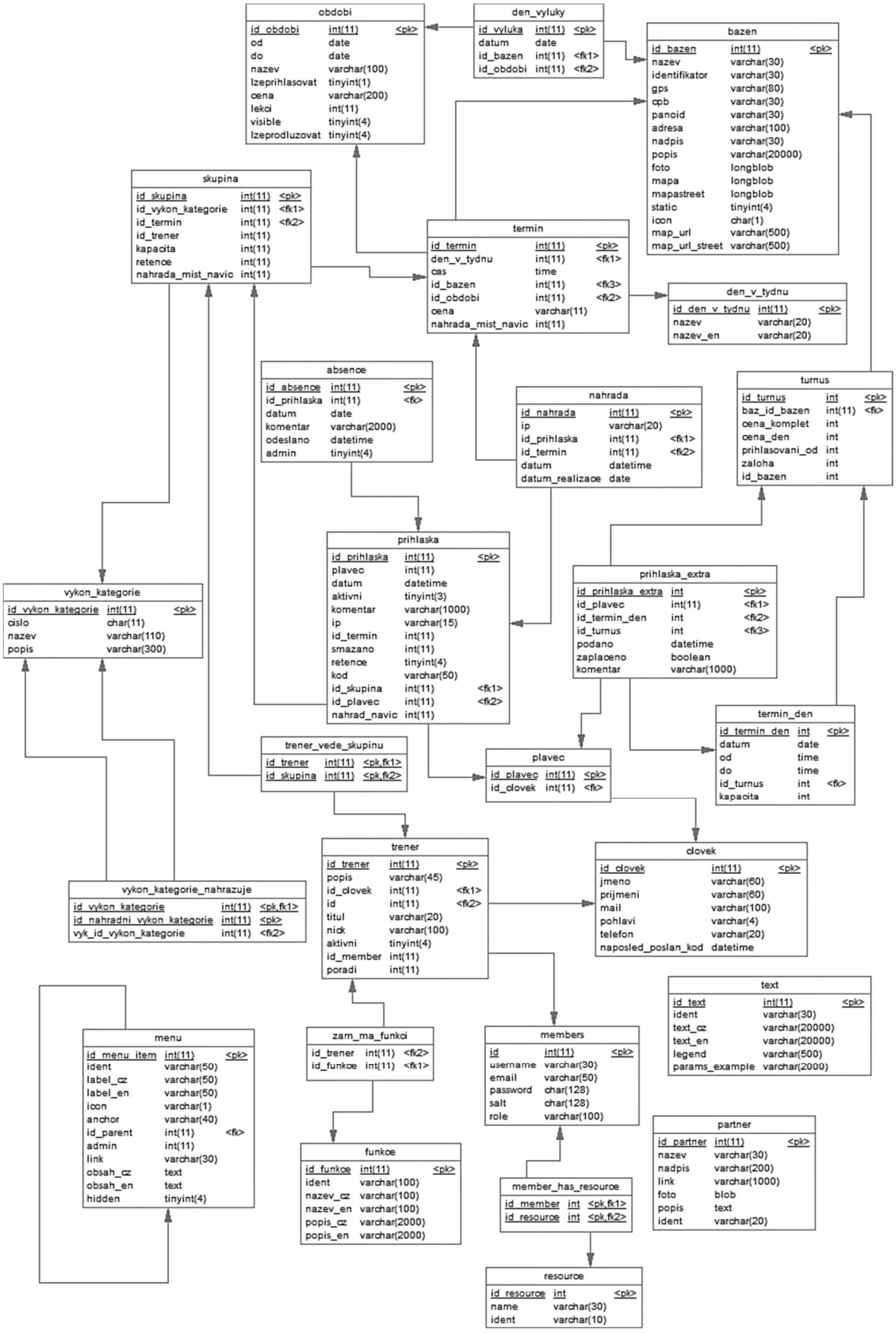
10.1.2 Reálná data

Reálná data (dále značená RD) použitá v této práci k výpočtu hodnot metrik a vybudování metadatového skladu jsou data vybraná z reálně existující aplikace; autor je jejím mnohaletým vývojářem i správcem. Slouží k organizování pravidelných sportovních kurzů pražskou soukromou firmou fungující od roku 2013. Zpracovaná data pocházejí ze dne 14. 1. 2019, pro názornou použitelnost v této práci byla vybrána jejich náhodná podmnožina (přibližně $\frac{1}{20}$, aniž by byla porušena referenční integrita) tak, aby schéma zůstalo zachováno. Data byla zcela anonymizována, takže citlivé osobní či firemní informace byly zcela vymazány nebo nahrazeny náhodnými. Navzdory zmenšení dat se jedná o rozsáhlý soubor, kdy anonymizovaná databáze má velikost 4,1MB. K výpočtům hodnot metrik a budování metadatového skladu v části 10.3 jsou následně použity vybrané podmnožiny zmenšených anonymizovaných dat.

Tato data jsou uložena zejména v relační databázi MySQL. Navzdory tomu značný podíl dat je nestrukturovaných a semistrukturovaných, odpovídajících textové reprezentaci řady formátů (zejména často založených na XML a JSON). Díky tomu se jedná o dostatečně heterogenní a ne zcela vhodně navrženou základnu pro různorodost výpočtu hodnot

všech tří metrik. Databázový fyzický model ukazuje Obrázek 25. Vzhledem k tomu, že je pracováno s již existujícími namodelovanými daty, není potřebné vytvářet zpětně konceptuální model, neboť k samotnému konceptuálnímu modelování již nedochází.

Dále pro účely sekce 10.3.4 zabývající se dlouhodobým monitorováním dat byly vybrány další **tři podmnožiny** těchto reálných dat. Jedná se o výběr stejných tabulek v databázi a jejich dat vztahujících se ke třem rokům – 2014, 2017 a 2020. V tomto období databáze procházela jistým vývojem, ale základní schéma zůstalo nezměněno.



Obrázek 25 – fyzické schéma relační databáze použitých reálných dat

10.2 Výpočet hodnot metrik – demonstrační data

Tato část popisuje výpočet hodnot metrik na základě demonstračních dat předvedených v části 10.1.1.

10.2.1 DD1 – relační databáze

Hodnoty metrik pro první demonstrační data DD1 jsou uspořádány do malého metadatového skladu manuálně v souladu s jeho modelem popsaným v části 8.2. Výsledný metadatový sklad ukazují následující Tabulka 8 a Tabulka 9.

Od pohledu je zřejmé, že míra strukturovanosti bude rovna 1 s jedinou výjimkou. Tou je sloupec `description` v tabulce `area` pro `id_area=1`. Definice hierarchičnosti je porušena u veškerých řádků tabulky `course`, majících dva cizí klíče. Porušení definice hierarchičnosti se ovšem neprojeví na hodnotě míry hierarchičnosti přímo řádků, ale na úrovni datového objektu, v rámci něhož k porušení dochází. Uvnitř něj se nachází kombinace vzájemně nehierarchicky odkazujících datových podobjektů a je jím v tomto případě hlavní datový objekt. Míra informace bude počítána algoritmickým způsobem (komprimační algoritmus bzip2).

object_id	parent_object_id	size	structuredness	hierarchicallity	information_amount	location_specification	existence_from	existence_to
1	NULL	201	0,8233	0,5278	0,8571	/	2019-03-01	NULL
2	1	99	0,6868	1	1	/area	2019-03-01	NULL
3	2	68	0,3546	1	1	/area/1	2019-03-01	NULL
4	2	13	1	1	1	/area/2	2019-03-01	NULL
5	2	17	1	1	1	/area/3	2019-03-01	NULL
6	3	13	1	1	1	/area/1/name	2019-03-01	NULL
7	3	54	0	1	1	/area/1/description	2019-03-01	NULL
8	4	12	1	1	1	/area/2/name	2019-03-01	NULL
9	4	0	1	1	1	/area/2/description	2019-03-01	NULL
10	5	11	1	1	1	/area/3/name	2019-03-01	NULL
11	5	5	1	1	1	/area/3/description	2019-03-01	NULL
12	1	42	1	1	1	/period	2019-03-01	NULL
13	12	21	1	1	1	/period/3	2019-03-01	NULL
14	12	21	1	1	1	/period/4	2019-03-01	NULL

15	13	10	1	1	1	/period/3/from	2019-03-01	NULL
16	13	10	1	1	1	/period/3/to	2019-03-01	NULL
17	14	10	1	1	1	/period/4/from	2019-03-01	NULL
18	14	10	1	1	1	/period/4/to	2019-03-01	NULL
19	1	60	1	1	1	/course	2019-03-01	NULL
20	19	12	1	1	1	/course/1	2019-03-01	NULL
21	19	12	1	1	1	/course/2	2019-03-01	NULL
22	19	12	1	1	1	/course/3	2019-03-01	NULL
23	19	12	1	1	1	/course/4	2019-03-01	NULL
24	19	12	1	1	1	/course/5	2019-03-01	NULL
25	20	1	1	1	1	/course/1/day	2019-03-01	NULL
26	20	8	1	1	1	/course/1/time	2019-03-01	NULL
27	21	1	1	1	1	/course/2/day	2019-03-01	NULL
28	21	8	1	1	1	/course/2/time	2019-03-01	NULL
29	22	1	1	1	1	/course/3/day	2019-03-01	NULL
30	22	8	1	1	1	/course/3/time	2019-03-01	NULL
31	23	1	1	1	1	/course/4/day	2019-03-01	NULL
32	23	8	1	1	1	/course/4/time	2019-03-01	NULL
33	24	1	1	1	1	/course/5/day	2019-03-01	NULL
34	24	8	1	1	1	/course/5/time	2019-03-01	NULL

Tabulka 8 – metadatový sklad (tabulka `data_object`, obsahující samotné objekty) prvních demonstračních dat DD1. Vzhledem k počtu sloupců bylo nutné zalomit některé obsahy buněk

referrer_id	**reffered_id**
20	3
20	13
21	5
21	13
22	5
22	13
23	3
23	14
24	4
24	14

Tabulka 9 - metadatový sklad (vazební tabulka `data_reference`, obsahující vzájemné odkazování řádků prostřednictvím cizích klíčů) prvních demonstračních dat DD1

Tento příklad ukazuje detailní princip budování metadatového skladu v šesti základních krocích.

Krok 1: **tvorba stromu datových objektů**, sloužící pro organizaci výpočtu metrik. Součástí je stanovení jejich ID, data jejich vytvoření a specifikace umístění – pro tuto jednoduchou RDBS je zvolen formát /tabulka/id/sloupec/. Při posloupnosti vytváření je vhodné dbát na to, aby při vkládání objektu byl již vložen jeho nadřazený objekt a nebylo tak nutné doplňovat dotyčný cizí klíč dodatečně.

Krok 2: **tvorba vazební tabulky** na základě vzájemného odkazování. To se v RDBS děje prakticky pouze na úrovni řádku.

Krok 3: **výpočet velikosti**. V nejjednodušších případech, jako je tento, lze za jednotku velikosti stanovit prostou textovou délku. Zde je podstatné, že velikost objektu nemusí být rovna součtu velikosti jeho podobjektů, neboť některé jsou z již zmíněných důvodů vynechány, avšak velikost je vždy počítána z celého objektu (v případě řádku tedy včetně klíčů, v případě tabulky včetně názvů sloupců apod.).

Krok 4: **výpočet míry hierarchičnosti**. Zde je podstatné, že je počítána vždy z konkrétního objektu samotného, izolovaného od okolí. Proto tedy míra hierarchičnosti řádků i tabulky `course` je rovna stále 1 – samotná tabulka hierarchičnost neporušuje. Způsobuje ovšem její porušení v kombinaci s oběma dalšími tabulkami, na něž odkazuje. Proto dotyčný (v tomto případě hlavní) datový objekt již zcela hierarchický není – jeho míra hierarchičnosti je rovna podílu (v závislosti na velikosti) veškerých jeho datových podobjektů, které v jeho kontextu neporušují hierarchii. Jedná se o všechny datové objekty kromě objektů lokalizovaných od `/course/1` po `/course/5`. Samotný výpočet je založen na elementárních podobjektech, které nejsou podřízené objektu porušujícímu definici hierarchičnosti, tedy ty s ID v rozsahu 6 – 11 a 15 – 18, jak stanovil již Vzorec 6. Je tedy použita modifikovaná podoba dělení zlomku sumou velikostí elementárních podobjektů, nikoli velikostí celého objektu, a to $\frac{\sum_{i=1}^{m} v(c_i^H)}{\sum_{j=1}^{n} v(c_j)}$. Pro hlavní objekt D_1 jde tedy o hodnotu $\frac{95}{180} = 0{,}5278$.

Z vypočtených hodnot míry hierarchičnosti tedy jiné vyplývá, že míra hierarchičnosti může být nižší než 1 u datového objektu odpovídajícímu jedné tabulce RDBS pouze v případě, že tato tabulka obsahuje minimálně dva rekurzivní (na primární klíč stejné tabulky odkazující) cizí klíče.

Krok 5: **výpočet míry strukturovanosti**. Zde platí princip, který popsal již Vzorec 4 – pro každý objekt, který není elementární (u nichž je hodnota stanovena přímo) je počítán vážený průměr z hodnoty, o kterou případný méně strukturovaný datový objekt míry strukturovanosti snižuje, násobené velikostním podílem, který dotyčný elementární objekt tvoří na celém objektu. Vzhledem k tomu, že se ve skladu nachází nestrukturovaný objekt 7 (`/area/1/description`), bude snížena míra strukturovanosti jeho nadřazeného objektu (řádku databáze, který jinak obsahuje druhý podobjekt 6, který je zcela strukturovaný) i dalších nadobjektů vyšších úrovní. Míra strukturovanosti objektu 3 je tedy vypočtena následujícím způsobem:

$$\frac{13\left(1-\frac{13}{68}(1-1)\right)+54\left(1-\frac{54}{68}(1-0)\right)}{68}=0{,}3546,$$

přičemž 13 je velikost podobjektu 6, dále 54 je velikost podobjektu 7 a 68 je velikost celého počítaného objektu 3. Podstatná je hodnota v závorce $\left(1-\mathrm{s}(\mathrm{c_j})\right)$ na levé straně čitatele. Zde podobjekt 6 je zcela strukturovaný a tedy míra, o kterou sníží celkovou míru strukturovanosti je tedy $1-1$, čili nulová.

Analogicky je postupováno u ostatních objektů nadřazených ne zcela strukturovaným podobjektům, tedy v tomto případě 2 a 1.

Krok 6: **výpočet míry informace**. Ten je v tomto případě založen na AIT a algoritmu bzip2. U kratších hodnot (zejména strukturovaných objektů nižší úrovně), jichž je v tomto příkladu většina, může velikost komprimovaného textového řetězce původní text i převýšit – v tomto případě je předpokládáno, že redundance neexistuje a míra informace je rovna 1. V opačném případě je počítán dosažený kompresní poměr prosté textové reprezentace datového objektu. Hlavní objekt je tedy reprezentován textem:

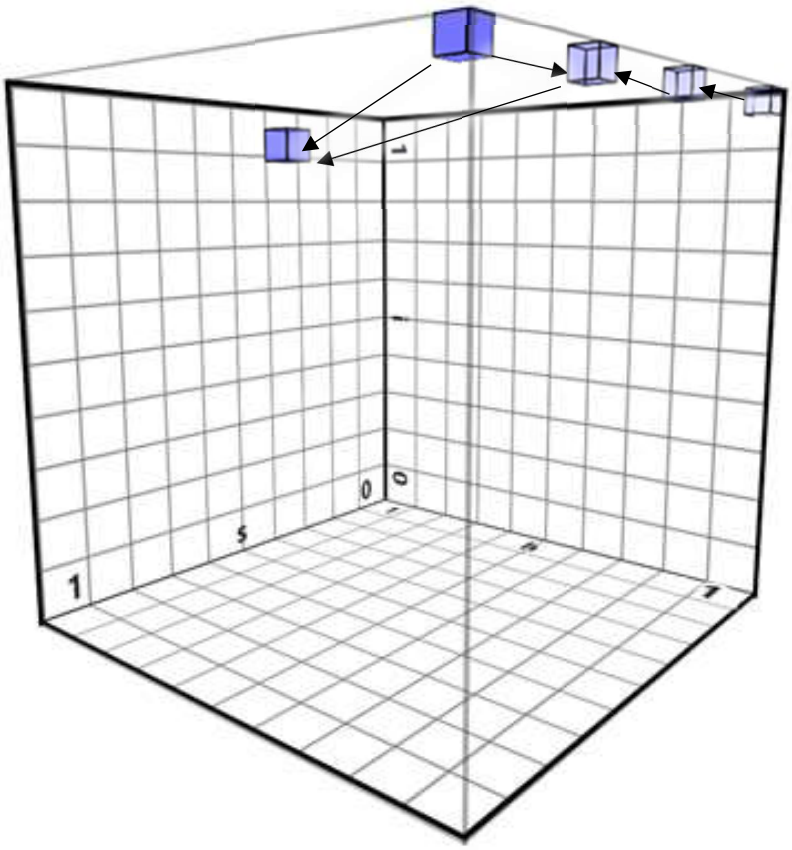
```
1 swimming pool This is an example long description of a swimming pool 2
football pit 3 tenis court Tenis 1 1 15:00:00 1 3 2 2 15:00:00 3 3 3 5
14:40:00 3 3 4 2 14:00:00 1 4 5 3 15:30:00 2 4 3 01.03.2019 30.06.2019 4
01.09.2018 26.02.2019
```

jehož komprimovaná podoba algoritmem bzip2 je následující:

```
eNpdjOsNwjAMhFe5CSrn0Yp2ji4QSgqR0jhqg2B8nCAVCfl+nL/zWeF4hW0L6Y7MHDE/wgGRS/B
vt+XoEVnCmz+WPeQSOIFXuL+axspcri5G5FBgUHySLws/94K5eVWnn4hEYo109A+YNj2UnewJbL
2wZ8VKbmrFNKAFGJDqyHSa1AhDHQ1faxsf63KBHjrSjX8AwA098g==
```

Výsledná míra informace je tedy rovna $\frac{204}{238}$ =0,8571.

Hodnoty metrik prvních demonstračních dat DD1 lze vizualizovat následujícím obrázkem:



Obrázek 26 – vizualizace prvních demonstračních dat DD1

Obrázek 26 vizualizuje oblasti trojrozměrného metadatového prostoru rozdělené po třech osách na rovnoměrné krychle o šířce $\frac{1}{20}$ celého prostoru. Průhlednost krychle je úměrná velikosti odpovídajících dat. Ačkoli data pokrývají nejširší škálu na ose míry strukturovanosti, je od pohledu názorné, že největší podíl tvoří vysoce až absolutně hierarchická, strukturovaná a neredundantní data. Šipky v tomto případě značí posloupnost oblastí pokrytých jednotlivými úrovněmi datových objektů směrem od nejnižší úrovně až k hlavnímu objektu. Prostor maximálních hodnot všech tří metrik obsahuje datové objekty více různých úrovní, proto od něj šipky vedou dvěma směry.

Míra kompatibility hlavního datového objektu D_1 a například obecné relační databáze (jejíž intervaly specifikovala Tabulka 2 v části 6.5) je rovna

$$k_{RDBS}(D_1) = 1 - \frac{1}{\sqrt{3}} \cdot \sqrt{(1 - 0{,}8233)^2 + (1 - 0{,}8571)^2} = 0{,}8688$$

Srovnání kompatibility datového objektu se všemi typy datových struktur ukazuje následující Tabulka 10.

Typ datové struktury	Kompatibilita
Relační databáze	0,8688
XML	0,7152
JSON	0,6974
RDF	0,8688
Nestrukturované formáty	0,4458

Tabulka 10 – kompatibilita hlavního datového objektu prvních demonstračních dat DD1 a obecných typů datových struktur

Výpočet míry kompatibility tedy zde ukázal poměrně očekávaný výsledek, kdy data účelově vytvořená pro RDBS mají vůči RDBS vyšší míru kompatibility než formáty XML a JSON, ovšem stejnou jako RDF. Hodnoty metrik zejména u objektů nižší úrovně jsou ale z naprosté většiny rovné 1 a tedy zcela kompatibilní se všemi 4 typy datových struktur (pochopitelně vyjma nestrukturovaných formátů).

Nižší výsledek kompatibility s RDBS je také do jisté míry ovlivněn výskytem jednoho nestrukturovaného datového objektu. Zde je třeba říct, že ačkoli Tabulka 2 pravdivě uvádí, že RDBS je určena především pro strukturovaná data, nečiní uložení takto malých nestrukturovaných dat žádný praktický problém a nebylo by případně chybou zpětně tabulku revidovat a rozšířit rozsah míry strukturovanosti z hodnoty {1} například na interval (0,9; 1⟩, tedy na data velmi, avšak ne nutně zcela strukturovaná.

10.2.2 DD2 – HTML stránka

U druhého příkladu, který tvoří fragment stránky ve formátu HTML je postupováno v krocích analogických k příkladu prvnímu; lze ovšem přepokládat, že vzhledem k odlišnosti typů datových struktur obou datových souborů nastanou velmi odlišné situace.

Krok 1: budování metadatového skladu prostřednictvím organizování **stromu datových objektů**, sloužícího pro výpočet hodnot metrik. Zde musí být vyřešena především specifikace umístění, neboť jednotlivé HTML elementy nemají unikátní identifikátory. Umístění je tak specifikováno umístěním elementu v hierarchii od kořenového HTML elementu až k dotyčnému – jejich názvy oddělené lomítky. Pokud existuje na stejné úrovni více elementů se stejným názvem, je do závorky za jménem umístěno číslo jeho pořadí. Atribut

elementu je adresován svým názvem v hranatých závorkách. Samotný textový uzel (v situaci, kdy se nachází ve smíšeném elementu soby sourozenec dalších elementů) je označen samotným číslem v závorce. Hlavním datovým objektem není samotný kořenový HTML element (který má svůj název tagu a případné atributy, což by znepřesnilo výpočet, navíc se jedná o fragment a na první úrovni může být elementů i více), avšak celý dokument.

Krok 2: **vazební tabulka**, není zde proveden – je zřejmé, že se jedná o zcela hierarchickou strukturu a data v této tabulce by zcela odpovídala hierarchii nadřazenosti samotných objektů.

Krok 3: **výpočet velikosti**. Zde je podstatné, že velikost elementu je vždy měřena z délky jeho vnitřního obsahu (ohrazeného párovými tagy), včetně prázdných znaků. Atributy a samotné tagy jsou vynechány (spadají do nadřazeného datového objektu). Velikost atributu je rovná délce jeho obsahu, bez názvu a uvozovek. Velikost textového uzlu je veškerá délka textového řetězce mezi sousedícími tagy, včetně případných okrajových mezer.

Krok 4: **výpočet míry hierarchičnosti**, je vynechán, neboť vybraný formát dat neumožňuje porušení hierarchie a míra hierarchičnosti je tedy rovna jedné.

Krok 5: **výpočet míry strukturovanosti**. Ten je počítán analogicky s předchozím příkladem, jak stanovuje Vzorec 4. Vzhledem k četnému podílu ne zcela strukturovaných dat nicméně výsledky jsou podstatně heterogennější. Zkoumaný objekt je vždy zkoumán na základě podřízených elementárních objektů, kterými jsou atomické elementy, textové uzly (v případě smíšených obsahů) a atributy.

Krok 6: je **výpočet míry informace**. Zde platí, že text, pro nějž je počítán kompresní poměr určující míru informace je shodný s textem, jehož délka byla měřena za účelem výpočtu velikosti ve třetím kroku. Zejména u menších objektů nižší úrovně opět nastává situace, kdy kompresí nelze velikost zmenšit a míra informace je tak rovna jedné. U nadobjektu zpravidla míra informace je výrazně nižší v momentě, kdy obsahuje vzájemně redundantní položky nebo jejich části – toto je zřetelně vidět například u objektu 33.

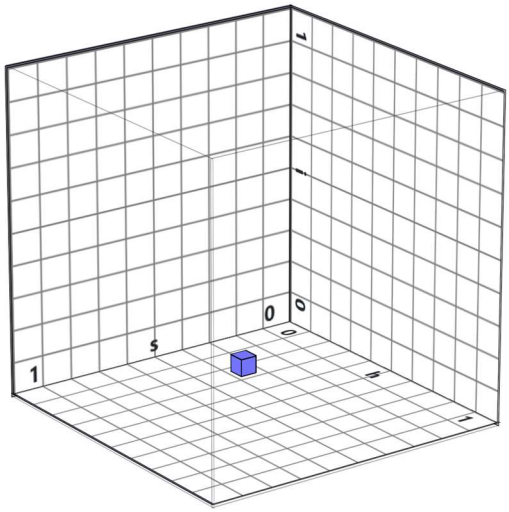
object_id	parent_object_id	size	structuredness	hierarchicallity	information_amount	location_specification
1	NULL	2733	0,8635	1,0000	0,3981	/
2	1	2697	0,8750	1,0000	0,3945	div
3	2	16	1,0000	1,0000	1,0000	div[class]
4	2	1244	0,4064	1,0000	0,3928	div/table(1)
5	4	19	1,0000	1,0000	1,0000	div/table(1)[class]
6	4	13	1,0000	1,0000	1,0000	div/table(1)[style]

7	4	0	1,0000	1,0000	1,0000	div/table(1)/thead
8	4	1121	0,4070	1,0000	0,3972	div/table(1)/tbody
9	8	378	0,5006	1,0000	0,5333	div/table(1)/tbody/tr(1)
10	9	24	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(1)
11	10	28	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(1)[style]
12	10	7	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(1)/strong
13	9	12	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(2)
14	13	28	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(2)[style]
15	9	1	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(3)
16	15	28	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(3)[style]
17	9	12	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(4)
18	17	28	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(4)[style]
19	9	18	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(5)
20	19	28	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(1)/td(5)[style]
21	8	362	0,3853	1,0000	0,5635	div/table(1)/tbody/tr(2)
22	21	23	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(1)
23	22	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(1)[style]
24	22	6	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(1)/strong
25	21	12	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(2)
26	25	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(2)[style]
27	21	1	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(3)
28	27	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(3)[style]
29	21	14	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(4)
30	29	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(4)[style]
31	21	45	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(5)
32	31	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(2)/td(5)[style]
33	8	306	0,3971	1,0000	0,4575	div/table(1)/tbody/tr(3)
34	33	0	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(1)
35	34	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(1)[style]
36	34	12	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(2)
37	36	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(2)[style]
38	33	1	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(3)
39	38	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(3)[style]
40	33	13	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(4)
41	40	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(4)[style]
42	33	12	0,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(5)
43	42	17	1,0000	1,0000	1,0000	div/table(1)/tbody/tr(3)/td(5)[style]
44	2	0	1,0000	1,0000	1,0000	div/hr(1)
45	2	276	0,7038	1,0000	1,0000	div/p(1)
46	45	20	1,0000	1,0000	1,0000	div/p(1)[style]
47	45	10	1,0000	1,0000	1,0000	div/p(1)/strong(1)
48	45	91	0,0000	1,0000	1,0000	div/p(1)/(1)
49	45	63	0,0000	1,0000	1,0000	div/p(1)/strong(2)
50	45	75	0,0000	1,0000	1,0000	div/p(1)/(2)
50	2	0	1,0000	1,0000	1,0000	div/hr(2)
51	2	990	0,7803	1,0000	0,4283	div/table(2)
52	51	28	1,0000	1,0000	1,0000	div/table(2)[style]
53	51	1	1,0000	1,0000	1,0000	div/table(2)[cellspacing]
54	51	1	1,0000	1,0000	1,0000	div/table(2)[cellpadding]
55	51	1	1,0000	1,0000	1,0000	div/table(2)[border]
56	51	5	1,0000	1,0000	1,0000	div/table(2)[class]
57	51	932	0,3866	1,0000	0,4249	div/table(2)/thead
58	57	890	0,4046	1,0000	0,4180	div/table(2)/thead/tr
59	58	3	1,0000	1,0000	1,0000	div/table(2)/thead/tr[valign]
60	58	344	0,4332	1,0000	0,6744	div/table(2)/thead/tr/th(1)
61	60	7	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)[bgcolor]
62	60	3	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)[width]
63	60	20	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/strong

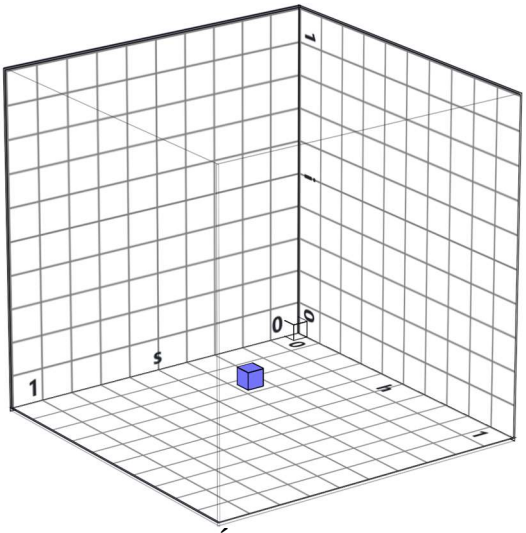
64	60	0	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/br
65	60	41	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/(1)
66	60	0	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/p
67	60	215	0,3798	1,0000	0,6884	div/table(2)/thead/tr/th(1)/ul
68	67	4	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/ul[type]
69	67	16	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/ul/li(1)
70	67	20	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/ul/li(2)
71	67	60	0,7322	1,0000	1,0000	div/table(2)/thead/tr/th(1)/ul/li(3)
72	71	8	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/ul/li(3)/(1)
73	71	15	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/ul/li(3)/a
74	73	22	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(1)/ul/li(3)/a[href]
75	58	386	0,3374	1,0000	0,6425	div/table(2)/thead/tr/th(2)
76	75	7	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)[bgcolor]
77	75	3	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)[width]
78	75	14	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/strong
79	75	0	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/br
80	75	12	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/(1)
81	75	0	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/p
82	75	227	0,4133	1,0000	0,6872	div/table(2)/thead/tr/th(2)/ul
83	82	4	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/ul[type]
84	82	16	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/ul/li(1)
85	82	20	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/ul/li(2)
86	82	72	0,7793	1,0000	1,0000	div/table(2)/thead/tr/th(2)/ul/li(3)
87	86	8	0,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/ul/li(3)/(1)
88	86	21	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/ul/li(3)/a
89	88	28	1,0000	1,0000	1,0000	div/table(2)/thead/tr/th(2)/ul/li(3)/a[href]
90	51	8	1,0000	1,0000	1,0000	div/table(2)/tbody
91	2	6	1,0000	1,0000	1,0000	div/p(1)

Tabulka 11 – metadatový sklad druhých demonstračních dat DD2. Poslední dva sloupce byly vynechány pro nedostatek místa a naprostou shodu s Tabulka 9

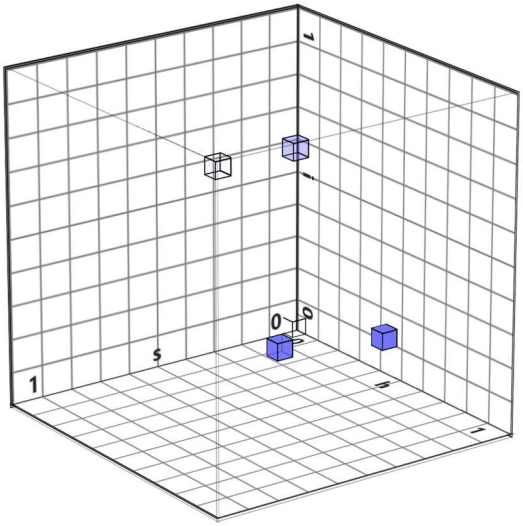
Metadatový sklad vytvořený na základě druhých demonstračních dat DD2 je opět možné vizualizovat. Výsledek tentokrát vzhledem k naprosté hierarchičnosti pokrývá pomyslnou pravou stěnu trojrozměrného metadatového prostoru, kde od pohledu tvoří jistý shluk přibližně kolem hodnoty 0,5 pro míru informace i míru strukturovanosti. Kompatibilita s formátem XML (ačkoli formát HTML mu teoreticky nemusí zcela odpovídat) je absolutní, jak ukazuje i Obrázek 11 a Obrázek 27 při vzájemném srovnání.



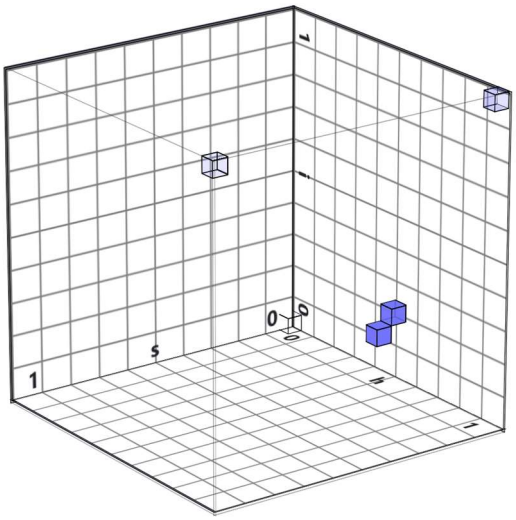
Úroveň 0 – hlavní datový objekt



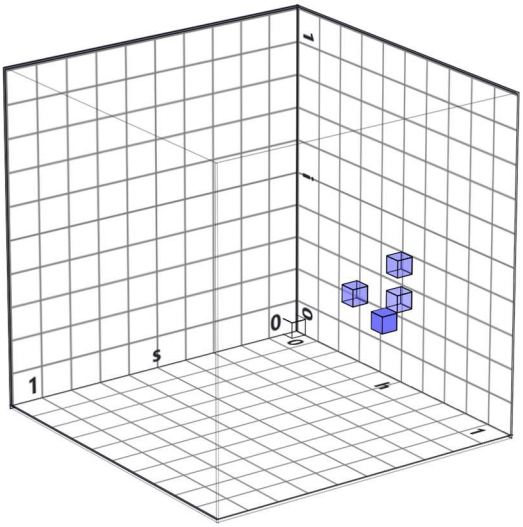
Úroveň 1



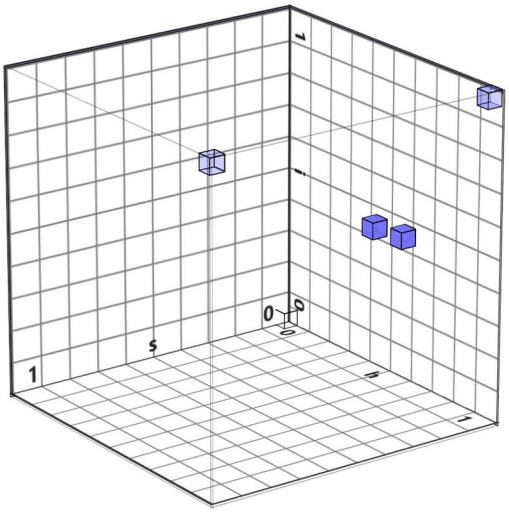
Úroveň 2



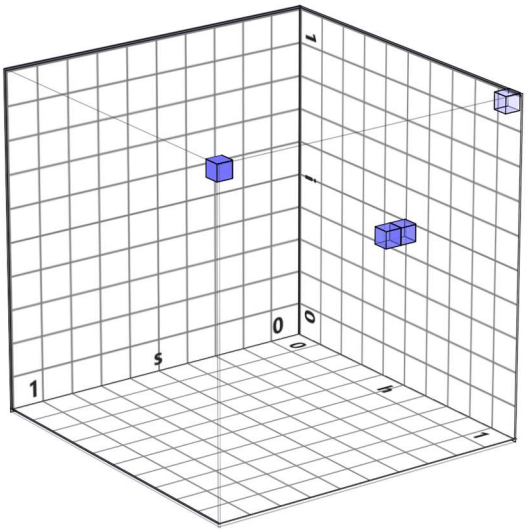
Úroveň 3



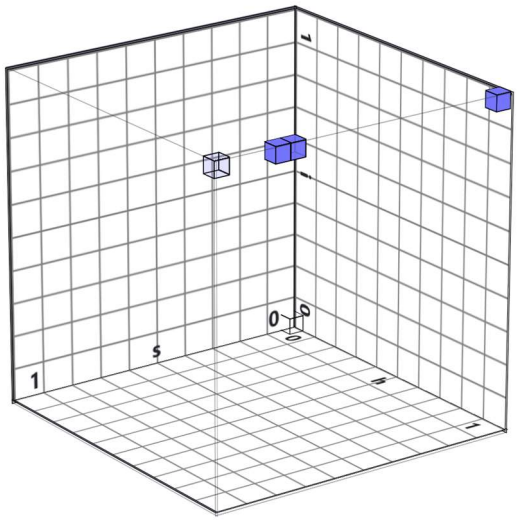
Úroveň 4



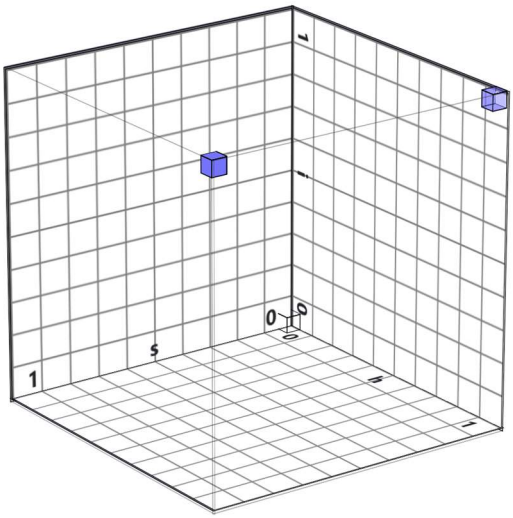
Úroveň 5



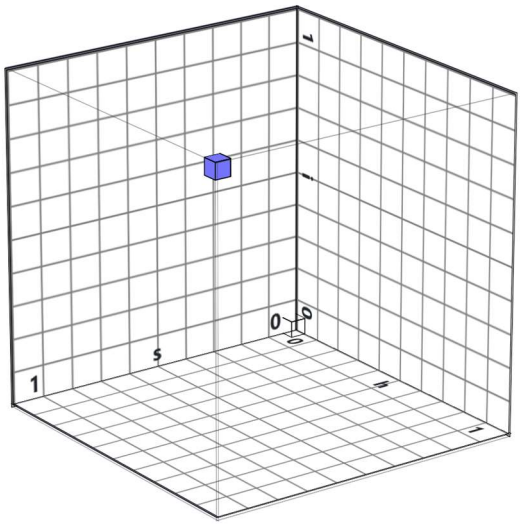
Úroveň 6



Úroveň 7



Úroveň 8



Úroveň 9

Obrázek 27 – vizualizace druhých demonstračních dat DD2 na deseti úrovních XML dokumentu

Obrázek 27 vizualizuje postupně všech 10 úrovní XML dokumentu, od hlavního datového objektu vlevo nahoře. Srovnání kompatibility datového objektu se všemi typy datových struktur ukazuje následující Tabulka 10.

Typ datové struktury	Kompatibilita
Relační databáze	0,6437
XML	1,0000
JSON	0,9212
RDF	0,9212
Nestrukturované formáty	0,5015

Tabulka 12 – kompatibilita hlavního datového objektu druhých demonstračních dat DD2 a obecných typů datových struktur

Výsledek dle očekávání ukázal nepříliš vysokou míru kompatibility s relační databází, zapříčiněnou jak nižšími hodnotami míry strukturovanosti, tak zejména velmi nízkou mírou informace. Nižší míra strukturovanosti měla vliv i na jisté snížení kompatibility s formáty JSON a RDF – u nich míra vyšla stejně z toho důvodu, že se liší pouze na ose míry hierarchičnosti (dle Tabulka 2), která je zde rovna 1. S nestrukturovanými formáty je kompatibilita ještě výrazně nižší – míra strukturovanosti není blízká nule.

10.2.3 DD3 – JSON objekt

Třetí příklad sestává z dat reprezentujících stejnou informaci jako příklad první, nicméně ve formátu JSON. Je postupováno opět v krocích, které se naopak do značné míry podobají druhému příkladu vzhledem k jisté podobnosti formátů JSON a XML.

Krok 1: tvorba **stromu datových objektů**, sloužící pro organizaci výpočtu metrik. Specifikace umístění je tentokrát poměrně jednoznačná, protože většina datových objektů má své pojmenování, které je na dané úrovni hierarchie unikátní. Jedinou výjimkou jsou pole – sekvence nepojmenovaných textových řetězců v hranatých závorkách oddělené čárkami. Pojmenování tedy sestává ze jmen podobjektů nebo proměnných pomocí cesty k jejich umístění oddělených čárkami, přičemž položka pole je adresována očíslováním dané položky v hranatých závorkách.

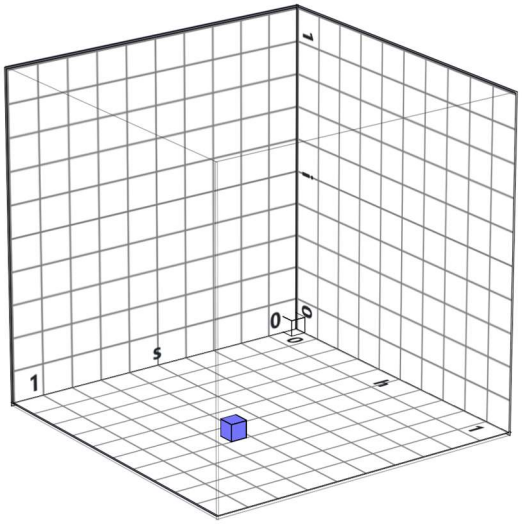
Krok 2: **vazební tabulka**, opět vynechán pro absolutní míru hierarchičnosti.

Krok 3: **výpočet velikosti**. Zkoumán je veškerý textový řetězec (v případě formátu JSON vymezený závorkami nebo uvozovkami) odpovídající objektu, a to včetně okrajových mezer.

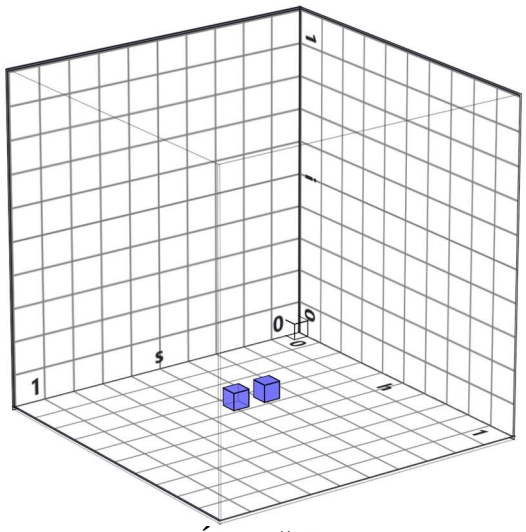
Krok 4: **výpočet míry hierarchičnosti** – je opět vynechán pro absenci porušení definice hierarchičnosti.

Krok 5: **výpočet míry strukturovanosti**. Podíl elementárního podobjektu je počítán dělením své velikosti a sumy velikosti všech elementárních podobjektů – toto pojetí nastínila již část 5.2. Důvodem je specifikum formátovaného textu reprezentujícího data ve formátu JSON, obsahující vysoký podíl prázdných znaků (zde výhradně mezer), a to i oproti formátu XML. Konkrétně zkoumaný XML dokument obsahuje 42,16% mezer, zatímco JSON 56,75%. Podíl datových podobjektů, mezi které mezery nejsou zahrnovány, je tak velmi malý. Při výpočtu se ukázalo, že míra strukturovanosti je přítomností nestrukturovaných datových podobjektů snížena zcela minimálně – v řádu jednotek procent. Tak je tomu u objektů tvořených z naprosté většiny nestrukturovaným datovým podobjektem. Například u hlavního objektu šlo původně o hodnotu 0,9876, tedy o hodnotu na první pohled vzhledem k nemalému podílu nestrukturovaných dat nesmyslnou.

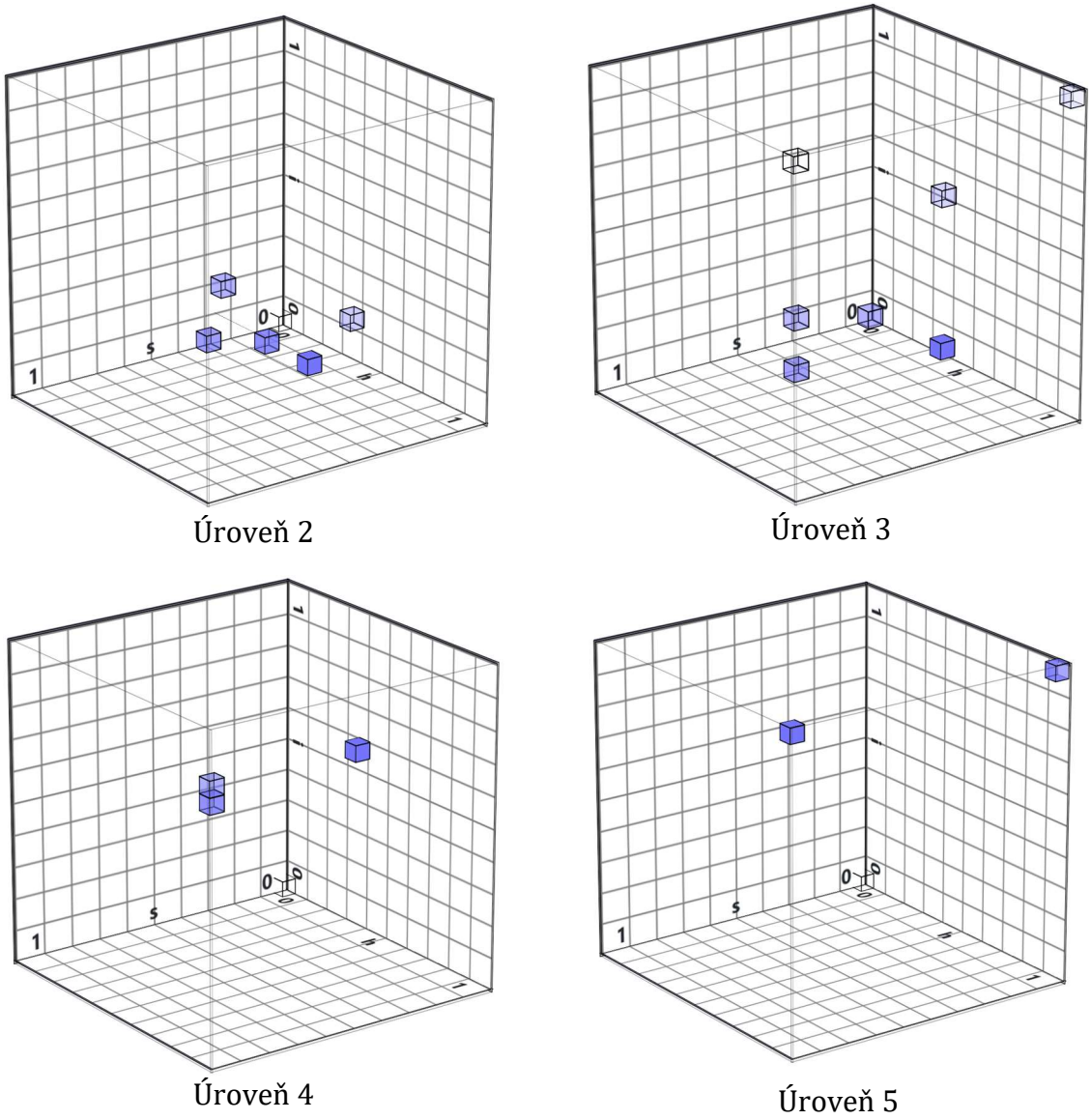
Krok 6: **výpočet míry informace**. Byla opět využita AIT a to na celém textovém řetězci reprezentujícím objekt včetně mezer.



Úroveň 0 – hlavní datový objekt



Úroveň 1



Obrázek 28 – vizualizace třetích demonstračních dat DD3 na šesti úrovních metadatového skladu

Celkovou vizualizaci ukazuje Obrázek 28. Výsledek se výrazně podobá předchozímu příkladu (DD2), zejména vzhledem k naprosté hierarchičnosti (rovna jedné). Je zde ovšem výraznější podíl dat na nižší úrovni míry informace, tedy pravděpodobně redundatních, ačkoli zcela různě strukturovaných.

Typ datové struktury	Kompatibilita
Relační databáze	0,5614
XML	1,0000
JSON	0,9661
RDF	0,9661
Nestrukturované formáty	0,4566

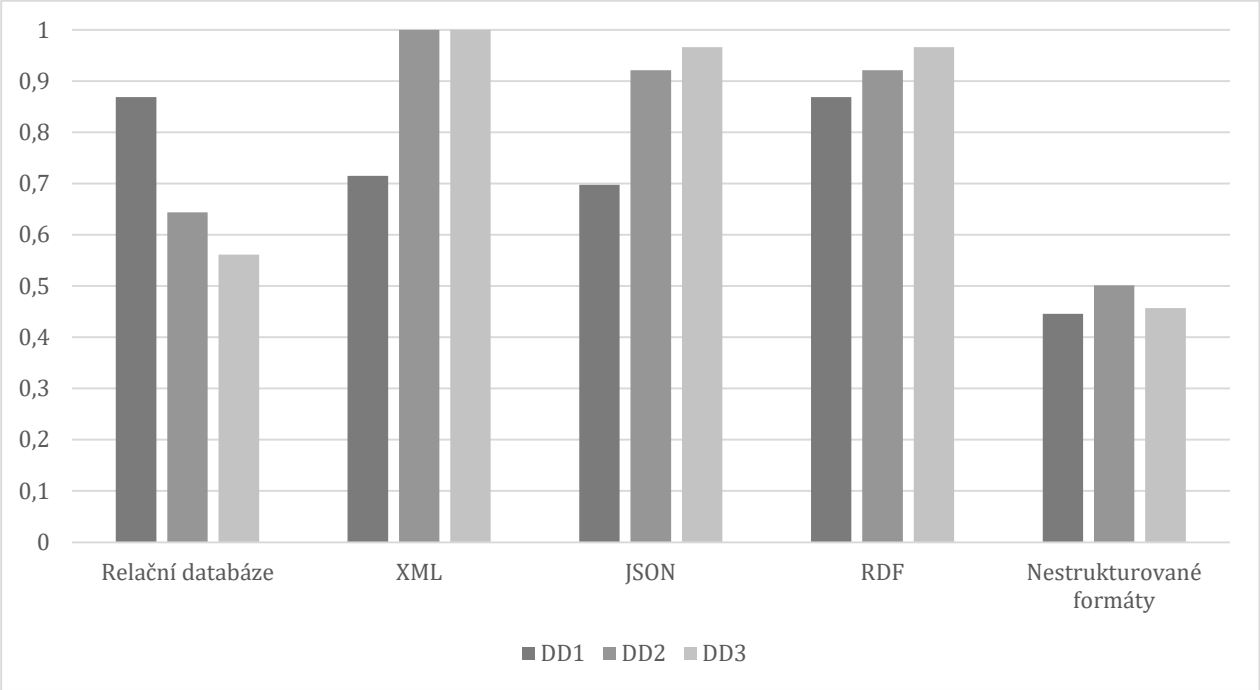
Tabulka 13 – kompatibilita hlavního datového objektu třetích demonstračních dat DD3 a obecných typů datových struktur

Kompatibilita, jejíž výsledky ukazuje Tabulka 13, ukazuje poměrně pochopitelnou nižší hodnotu pro relační databázi vzhledem k nízké míře informace silně redundantních dat. Navzdory formátu demonstračních dat JSON vyšla naprostá kompatibilita (rovna jedné) s formátem XML, a to pravděpodobně vzhledem k míře strukturovanosti, která je u formátu XML libovolná, zatímco u JSON a RDF vedla u obou typů datových struktur ke shodnému poklesu hodnoty kompatibility. Vzhledem k vysoké míře strukturovanosti také vyšla nižší kompatibilita s nestrukturovanými formáty.

object_id	parent_object_id	size	structuredness	hierarchicality	information_amount	location_specification
1	NULL	1764	0,9412	1,0000	0,2426	/
2	1	961	0,8498	1,0000	0,3330	coursesByArea
3	2	934	0,6348	1,0000	0,3298	coursesByArea/swimming_pool
4	3	54	0,0000	1,0000	1,0000	coursesByArea/swimming_pool/description
5	3	827	0,4898	1,0000	0,3047	coursesByArea/swimming_pool/courses
6	5	114	0,4898	1,0000	0,8070	coursesByArea/swimming_pool/courses[1]
7	6	1	1,0000	1,0000	1,0000	coursesByArea/swimming_pool/courses[1]/day
8	6	5	1,0000	1,0000	1,0000	coursesByArea/swimming_pool/courses[1]/time
9	6	15	0,0000	1,0000	1,0000	coursesByArea/swimming_pool/courses[1]/period
10	5	114	0,4898	1,0000	0,8421	coursesByArea/swimming_pool/courses[2]
11	10	1	1,0000	1,0000	1,0000	coursesByArea/swimming_pool/courses[2]/day
12	10	5	1,0000	1,0000	1,0000	coursesByArea/swimming_pool/courses[2]/time
13	10	15	0,0000	1,0000	1,0000	coursesByArea/swimming_pool/courses[2]/period
14	2	190	0,4898	1,0000	0,4246	coursesByArea/football_pit
15	14	0	1,0000	1,0000	1,0000	coursesByArea/football_pit/description
16	14	137	0,4898	1,0000	0,7591	coursesByArea/football_pit/courses
17	16	114	0,4898	1,0000	0,8421	coursesByArea/football_pit/courses[1]
18	17	1	1,0000	1,0000	1,0000	coursesByArea/football_pit/courses[1]/day
19	17	5	1,0000	1,0000	1,0000	coursesByArea/football_pit/courses[1]/time
20	17	15	0,0000	1,0000	1,0000	coursesByArea/football_pit/courses[1]/period
21	2	537	0,7963	1,0000	0,4246	coursesByArea/tenis_court
22	21	5	1,0000	1,0000	1,0000	coursesByArea/tenis_court/description
23	21	266	0,7449	1,0000	0,4511	coursesByArea/tenis_court/courses
24	23	114	0,4898	1,0000	0,8070	coursesByArea/tenis_court/courses[1]
25	24	1	1,0000	1,0000	1,0000	coursesByArea/tenis_court/courses[1]/day
26	24	5	1,0000	1,0000	1,0000	coursesByArea/tenis_court/courses[1]/time
27	24	15	0,0000	1,0000	1,0000	coursesByArea/tenis_court/courses[1]/period
28	23	114	0,4898	1,0000	0,8070	coursesByArea/tenis_court/courses[2]
29	28	1	1,0000	1,0000	1,0000	coursesByArea/tenis_court/courses[2]/day
30	28	5	1,0000	1,0000	1,0000	coursesByArea/tenis_court/courses[2]/time
31	28	15	0,0000	1,0000	1,0000	coursesByArea/tenis_court/courses[2]/period
32	1	756	0,9449	1,0000	0,3386	coursesByPeriod
33	32	425	0,9513	1,0000	0,4518	coursesByPeriod[1]
34	33	15	0,0000	1,0000	1,0000	coursesByPeriod[1]/range
35	33	355	1,0000	1,0000	0,3718	coursesByPeriod[1]/courses
36	35	102	1,0000	1,0000	0,7843	coursesByPeriod[1]/courses[1]
37	36	1	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[1]/day
38	36	5	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[1]/time
39	36	13	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[1]/area
40	35	100	1,0000	1,0000	0,8000	coursesByPeriod[1]/courses[2]
41	40	1	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[2]/day
42	40	5	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[2]/time
43	40	11	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[2]/area
44	35	100	1,0000	1,0000	0,8000	coursesByPeriod[1]/courses[3]
45	44	1	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[3]/day
46	44	5	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[3]/time
47	44	11	1,0000	1,0000	1,0000	coursesByPeriod[1]/courses[3]/area
48	32	311	0,9168	1,0000	0,6045	coursesByPeriod[2]
49	48	15	0,0000	1,0000	1,0000	coursesByPeriod[2]/range
50	48	241	1,0000	1,0000	0,5311	coursesByPeriod[2]/courses
51	50	102	1,0000	1,0000	0,7843	coursesByPeriod[2]/courses[1]
52	51	1	1,0000	1,0000	1,0000	coursesByPeriod[2]/courses[1]/day
53	51	5	1,0000	1,0000	1,0000	coursesByPeriod[2]/courses[1]/time
54	51	13	1,0000	1,0000	1,0000	coursesByPeriod[2]/courses[1]/area
55	50	101	1,0000	1,0000	0,7921	coursesByPeriod[2]/courses[2]
56	55	1	1,0000	1,0000	1,0000	coursesByPeriod[2]/courses[2]/day
57	55	5	1,0000	1,0000	1,0000	coursesByPeriod[2]/courses[2]/time
58	55	12	1,0000	1,0000	1,0000	coursesByPeriod[2]/courses[2]/area

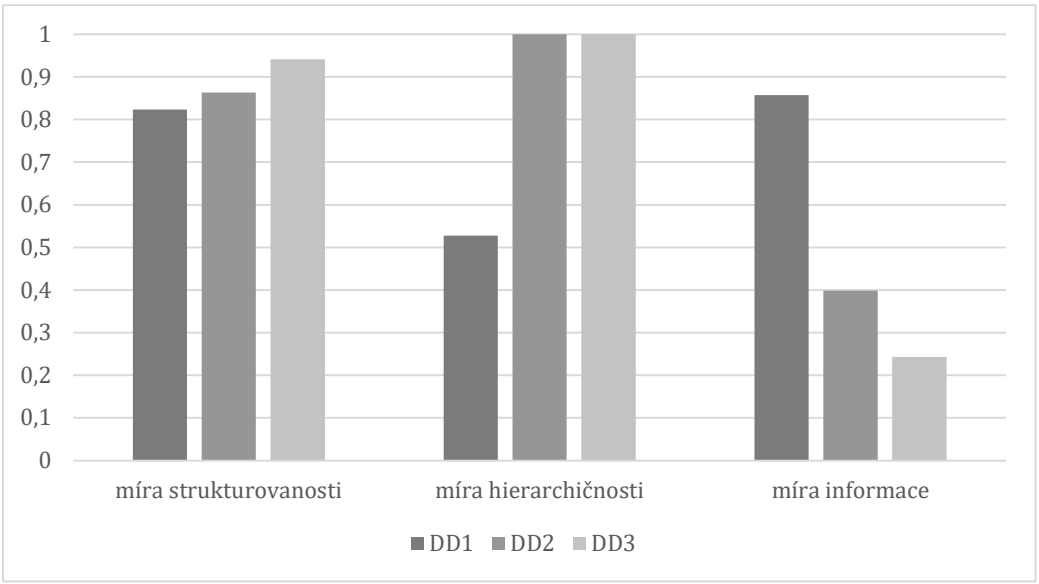
Tabulka 14 – metadatový sklad třetích demonstračních dat DD3

10.2.4 Srovnání demonstračních dat



Obrázek 29 – kompatibilita demonstračních dat s typy datových struktur

Jak ukazují Obrázek 29 a Obrázek 29, první data se v hodnotách kompatibility výrazně liší od druhých a třetích, která se navzájem poměrně shodují. Je také vidět, že kompatibilita všech tří dat s formátem XML a JSON je pro oba formáty podobná, ale především pro relační databázi se velmi liší.



Obrázek 30 – hodnoty metrik tří demonstračních dat

10.2.5 Datová transformace

Tato transformace využívá výsledky ze sekcí 10.2.1 a 10.2.3. Oboje data – RDBS DD1a JSON dokument DD3 jsou reprezentanty stejné informace a je možné je tak chápat jako **počáteční a koncový stav** transformace, v tomto případě od DD1 k DD3.

Analyzujeme-li transformaci coby převod dat RDBS na objekt JSON (což je velmi běžný proces například v oblasti webových aplikací), dochází k porovnání hodnot hlavních datových objektů obou metadatových skladů.

object_id	parent_object_id	size	structuredness	hierarchicallity	information_amount	location_specification
D_0	NULL	201	0,8233	0,8967	0,8571	rdbs/
D_1	NULL	1764	0,9412	1,0000	0,2426	json/

Tabulka 15 – srovnání hodnot na začátku a konci procesu datové transformace

Tabulka 15 porovnává dotyčné hodnoty. Jedná se o dva datové objekty pomyslného dalšího metadatového skladu (lze předpokládat, že v praxi by objekty z tabulek Tabulka 8 a Tabulka 14 byly umístěné v rámci jednoho metadatového skladu), proto neodpovídají primární klíče a specifikace umístění a datové rozsahy byly vynechány.

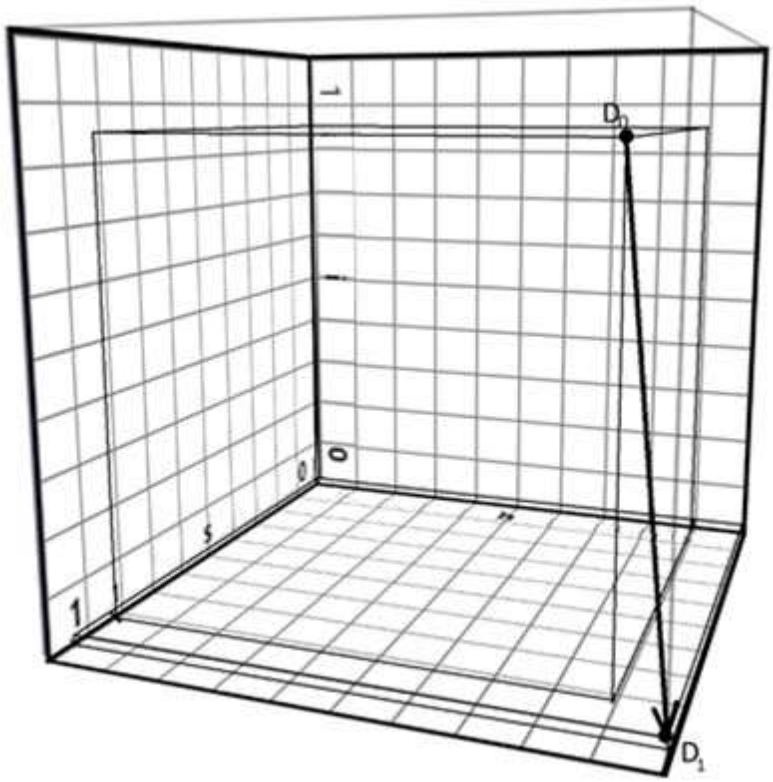
Na základě pohybů hodnot metrik je možné okamžitě transformaci klasifikovat coby mírnou strukturalizaci, mírnou hierarchizaci a denormalizaci.

$$\begin{aligned}&\big(s(DD3)-s(DD1), h(DD3)-h(DD1), i(DD3)-i(DD1)\big)=\\&=(0{,}9412-0{,}8233;\ 1-0{,}8967; 0{,}2426-0{,}8971)=\\&=(0{,}1179; 0{,}1033; -0{,}6145)\end{aligned}$$

Vzorec 15 – transformační vektor převodu RDBS na objekt JSON

Vzorec 15 ukazuje výpočet transformačního vektoru způsobem, jaký stanovil Vzorec 12. Z něj může být spočítána celková délka transformace:

$$t(DD1, DD3)=\frac{1}{\sqrt{3}}\cdot\sqrt{0{,}1179^2+0{,}1033^2+(-0{,}6145)^2}=0{,}3661.$$



Obrázek 31 – vizualizace datové transformace v trojrozměrném metadatovém prostoru

Obrázek 31 znázorňuje vizualizaci transformace. Jsou doplněny vodící linky pro představu reálné pozice bodů D_0 a D_1, protože tato práce neumožňuje vložení vizuálně vhodnější trojrozměrné animace rotujícího trojrozměrného metadatového prostoru.

Ještě je vhodné připomenout, že původní **kompatibilita** DD1 vůči formátu JSON byla spočítána jako 0,6974 (uvádí Tabulka 10), což znamená, že vzdálenost DD1 od typu datové struktury JSON je rovna $1 - 0{,}6974 = 0{,}3026$. Tato hodnota je podobná jako výsledná délka transformace DD1 na DD3, která je rovna 0,3661. Transformační proces tedy proběhl dle očekávání.

10.3 Výpočet hodnot metrik – reálná data

Ukázkový výpočet hodnot metrik na datové struktuře vychází z reálných dat (dále značených RD) popsaných v sekci 10.1.2. Vybrány jsou dvě maximálně heterogenní podmnožiny dotyčné relační databáze.

Prvními reálnými daty RD1 jsou klientské rezervace v posledním období a veškeré na ně navázané hodnoty – klienti, vypsané termíny, areály a dotyčné období. Tuto podmnožinu lze obecně charakterizovat jako výrazně strukturovanou, málo redundantní a více, avšak ne zcela hierarchickou.

Druhými reálnými daty RD2 je tabulka `text`, obsahující parametrizované šablony pro generování HTML dokumentů pro zobrazení přímo na webu nebo použití v mailech. Tato podmnožina obecně je charakterizovatelná coby částečně strukturovaná až nestrukturovaná, velmi redundantní a zcela hierarchickou.

Vzájemná odlišnost obou reálných dat by opět měla zajistit, že ukázkový výpočet metrik pokryje co možná největší škálu možných situací, na které lze při výpočtu narazit.

Vstupem experta je v tomto procesu především zjištění, že databáze neobsahuje neformální vazby a pro výpočet míry hierarchičnosti postačí znalost cizích klíčů. Dále bylo nutné stanovit kritérium strukturovanosti u elementárních objektů; stejně jako navrhuje sekce 5.2 byly za nestrukturované pokládané ty, které obsahují textové mezery a lze tedy předpokládat, že jde o texty v přirozeném jazyce. V případě, že by znalost experta dotyčných dat potvrdila přítomnost textů, na které nelze toto jednoduché pravidlo aplikovat, bylo by nutné kritérium upravit.

10.3.1 RD1 – část podnikové databáze

Ze schématu, kterou znázornil již Obrázek 25, jsou do prvních reálných dat RD1 vybrány tabulky `obdobi`, `bazen`, `skupina`, `termin`, `prihlaska`, `plavec`, `clovek` a `vykon_kategorie`, včetně veškerých vzájemných relací. Pro jednoduchost jsou dále tříděna data výhradně dle položky `id_obdobi=12` a na ní vázaných záznamů. Takováto množina jako celek je datovým objektem.

Výpočet hodnot metrik bude ovšem z důvodu následného agregování probíhat v hierarchii objektů odspoda směrem nahoru z důvodu potřeby znát hodnoty elementárních podobjektů především při výpočtu míry strukturovanosti.

Nejdříve jsou počítány hodnoty metrik objektů jednotlivých buněk v řádcích RDBS, a to všech vyjma těch, jejichž charakter je čistě strukturální. Vynechány jsou proto opět primární a cizí klíče (mohly by být zachovány, pakliže by zároveň byly nositeli další informace, která již ale bude zachycena ve vazební tabulce referencí objektů). Míra strukturo-

vanosti je rovna jedné pro všechny položky kromě těch, obsahujících více slov, tedy řetězců oddělených mezerami, které budou pokládány za nestrukturované – vyskytují se ve více tabulkách pod sloupci s názvy `nazev`, `gps`, `adresa`, `nadpis`, `popis` a `komentar`. Veškeré prázdné (`null`) položky jsou vynechány. Míra hierarchičnosti je vždy 1 vzhledem k tomu, že buňka řádku nemůže odkazovat na nic jiného než svůj řádek a je tak vždy zcela hierarchická. Míra informace je spočítána coby podíl, který stanovil již Vzorec 12 za použití algoritmu bzip2.

Následně jsou zkoumány řádky tabulek. Protože řádek tabulky RDBS ukládá data (která samotná nemusí být strukturovaná) strukturovaným způsobem, je aplikován Vzorec 4; položky `null` jsou opět vynechány. Míra hierarchičnosti je již závislá na existenci cizího klíče – pakliže jsou více než dva (vazba řádku na tabulku není v tomto případě klasickou informační vazbou a může být ze stanovení hierarchičnosti vynechána), pak lze mluvit o porušení hierarchičnosti a její hodnota je tedy 0. Míra informace je spočítána analogickým způsobem jako u buňky, přičemž zkoumanými daty je prostý výpis neprázdných obsahů jednotlivých buněk oddělený řádky.

Celé tabulky agregují hodnoty metrik svých řádků velmi podobným způsobem jako předchozí typy databázových objektů. Z hlediska výpočtu metrik za účelem následného budování datového skladu je tabulka pouhou nadentitou zastřešující data o velmi podobném charakteru. Všechny hodnoty metrik jsou proto opět vypočteny z textové reprezentace všech jejich dat, popřípadě odpovídajících elementárních podobjektů. Stejný postup platí i pro datový objekt nejvyšší úrovně, jímž je celá databáze.

10.3.2 RD2 – šablony pro generování webového obsahu

Druhá reálná data RD2 mají vzhledem k charakteru uložené informace natolik odlišný charakter, že aplikovat zcela stejný postup jako u prvních reálných dat RD1 by znamenalo zbytečné opomenutí potenciálně zajímavých vnitřních charakteristik v buňkách se šablonami pro generování textů.

Tato data neobsahují cizí klíče a její míra hierarchičnosti (i vzhledem k vnitřnímu použití formátů XML a JSON) je tak rovna 1. Míra informace je spočítána stejným způsobem jako u prvních reálných dat RD1. Hlavní odlišností tak zůstává fakt, že buňky ve sloupci `text_cz`, `text_en` a `params_example` obsahují data ve formátech XML a JSON, která jsou ne zcela vhodně uložena jako prostý text. Uložení přímo jako XML a JSON databázový systém MySQL nepodporuje.

Hierarchie objektů u těchto dat má nicméně stejný počet úrovní jako v případě prvních reálných dat RD1. Vzhledem k textovému uložení tedy bude zacházeno i s těmito sloupci jako s prostým textem.

Ostatní sloupce tabulky RDBS již obsahují atomické obsahy a jsou organizovány do dalších objektů. Jim je podobně jako v předchozím případě nadřazen řádek tabulky a následně celá tabulka.

10.3.3 Sestavení metadatového skladu

Proces sestavení metadatového skladu (v této kapitole automatizovaným způsobem) je jistou analogií k ETL procesům používaným pro obdobnou činnost v oblasti klasických datových skladů. Je tedy přepokládáno jejich možné periodické spouštění na měnících se datech, ovšem neměnícím se schématu dat.

Datový sklad je sestavován především na základě SQL dotazů pro danou databázi – veškerá ukázková data jsou uložena právě v ní, jak popsala sekce 10.1.2. Tyto SQL dotazy jsou maximálně univerzální pro dané schéma.

Veškerý následující postup je prováděn paralelně pro oboje zkoumaná reálná data.

Východiskem pro uložení dat do metadatového skladu je samotná struktura metadatového skladu na základě schématu, které znázornil Obrázek 22, jejíž založení provádí Kód 3.

```
CREATE TABLE `data_object` (
  `object_id` int(11) NOT NULL,
  `parent_object_id` int(11) DEFAULT NULL,
  `size` int(11) DEFAULT NULL,
  `structuredness` float DEFAULT NULL,
  `hierarchicallity` float DEFAULT NULL,
  `information_amount` float DEFAULT NULL,
  `location_specification` text COLLATE utf8mb4_bin,
  `existence_from` datetime DEFAULT NULL,
  `existence_to` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;

CREATE TABLE `data_reference` (
  `referrer_id` int(11) NOT NULL,
  `reffered_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin;

ALTER TABLE `data_object`
  ADD PRIMARY KEY (`object_id`),
  ADD KEY `parent_object_id` (`parent_object_id`);

ALTER TABLE `data_reference`
  ADD KEY `referrer_id` (`referrer_id`,`reffered_id`),
```

```
  ADD KEY `referred` (`reffered_id`);


ALTER TABLE `data_object`
  MODIFY `object_id` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `data_object`
  ADD CONSTRAINT `parent` FOREIGN KEY (`parent_object_id`) REFERENCES
`data_object` (`object_id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `data_reference`
  ADD CONSTRAINT `referred` FOREIGN KEY (`reffered_id`) REFERENCES
`data_object` (`object_id`),
  ADD CONSTRAINT `referrer` FOREIGN KEY (`referrer_id`) REFERENCES
`data_object` (`object_id`);
ALTER TABLE `data_reference`
  ADD PRIMARY KEY( `referrer_id`, `reffered_id`);
```

Kód 3 – vytvoření metadatového skladu

Jazyk SQL nicméně pro vybudování metadatového skladu není zcela vhodný – neumožňuje vypisovat průběh velmi dlouhých transformačních procesů již během jejich konání, jeho ladění není příliš flexibilní a obtížně se kontrolují hodnoty datových typů. Proto je transformační script vytvořen v autorovi bližších technologiích založených na jazyce PHP 7.3 a knihovně Nette Database 3.0, běžících v uzavřeném virtualizovaném kontejneru na bázi Dockeru. Samotná databáze je systém MySQL 5.6.40.

Aplikace přistupuje ke dvěma databázím – zdrojové (obsahující zkoumaná data) a cílové (obsahující metadatový sklad). Samotný skript je plně ad-hoc, tedy není přímo znovupoužitelný na jakákoli data, avšak při opětovném spuštění zafunguje i na jiných datech o stejném schématu, čímž je možné i dlouhodobější monitorování.

Samotný postup prakticky odpovídá postupu uvedeném již v sekci 10.2.1, ovšem vzhledem k aplikačnímu řešení obsahuje řadu specifik.

Krok 1**:** opět budování metadatového skladu prostřednictvím organizování **stromu datových objektů**, počínaje hlavním datovým objektem. Zde je specifikováno umístění v databázi (znak „/“) a platnost (stanovena fixně od 1. 3. 2019 v 00:00). Následují jednotlivé tabulky, u kterých je pouze lokace umístění specifikována na název tabulky s lomítkem před i po názvu – takto specifikace navazuje na nadřazený, tedy hlavní objekt. Po tabulkách jsou iterovány a vkládány jednotlivé řádky; u nich je k umístění přidán dotyčný primární klíč a znak lomítka (složený primární klíč se v této databázi nevyskytuje).

Krok 2: **plnění vazební tabulky** `data_reference` vzájemného odkazování prostřednictvím cizích klíčů. Vstupem algoritmu je seznam tabulek, jejich cizích klíčů a tabulek, na

něž odkazují, kritériem pro vložení relace je, že hodnota cizího klíče není prázdná (v jazyce SQL značeno `IS NOT NULL`). I zde platí, že odkazujícím datovým objektem může být pouze ten na úrovni tabulky.

Krok 3**: výpočet velikostí** datových objektů. Protože databáze neobsahuje binární data (existující sloupce typu BLOB jsou prázdné), je počítána dle textové délky. Jediným problémem může být `datetime`, který je nutné pro textovou podobu naformátovat – je použit formát `%Y-%M-%D %H:%I:%S`, tedy například „2019-01-21 22:35:50“. V první řadě jsou počítány velikosti buněk, ty jsou následně sčítány pro řádky, tabulky i celý metadatový sklad.

Krok 4**:** výpočet **míry hierarchičnosti**. Tato hodnota má smysl pouze u hlavního datového objektu prvních reálných dat RD1 – jednotlivé tabulky samotné definici hierarchičnosti porušit nemohou kvůli neexistenci rekurzivních relací. Zde musí být v prvé řadě shromážděny objekty, které v kontextu hlavního datového objektu odkazují na více dalších objektů a porušují tak definici hierarchičnosti a spočtena jejich velikost, a to pomocí SQL kódu Kód 4.

```
SELECT referrer_id,
       Count(referrer_id) AS refference_count,
       Sum(data_object.size) AS object_size
FROM data_reference
LEFT JOIN data_object ON (data_reference.referrer_id = data_object.object_id)
GROUP  BY data_reference.referrer_id
HAVING Count(referrer_id) > 1
```

Kód 4 – identifikace datových objektů porušujících definici hierarchičnosti v kontextu hlavního objektu

Krok 5**:** výpočet **míry strukturovanosti**. Opět je postupováno od nejnižší úrovně (kritériem nestrukturovanosti elementárního datového objektu je výskyt více než jedné mezery v textovém řetězci), tedy buněk tabulky, přičemž sloupce s datem a časem jsou formátovány stejně jako ve třetím kroku. Výpočet je tak v plném souladu se sekcí 10.2.1.

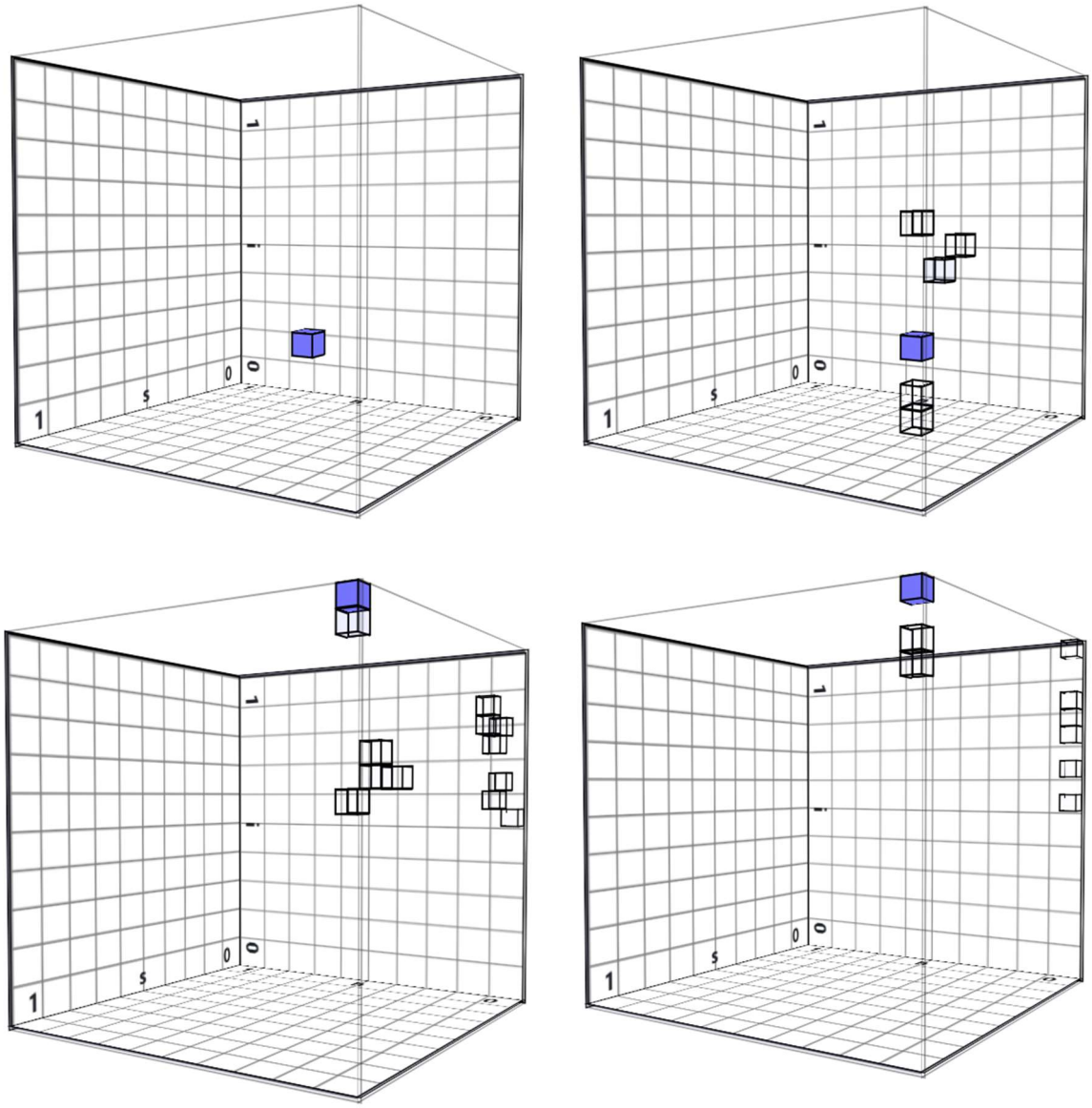
Krok 6: výpočet **míry informace**. Je užito textových řetězců, které u neelementárních datových objektů jsou jednoduše spojeny za sebou do jediného bez mezer. Pro výpočet kompresního poměru je užita vestavěná funkce jazyka php `gzencode`, přičemž její parametr `level` je nastaven na maximální hodnotu 9, tedy nejvyšší možná komprese odpovídající principům AIT.

Vzhledem k vysokému počtu datových objektů (25902 u prvních reálných dat RD1 a 562 u druhých reálných dat RD2) uvádí Tabulka 16 pouze datové objekty odpovídající celému metadatovému skladu a jednotlivým tabulkám a Tabulka 17 navíc i datové objekty odpovídající řádkům.

object_id	parent_object_id	size	structuredness	hierarchicallity	information_amount	location_specification	existence_from	existence_to
1	NULL	189540	0.999696	0.876248	0.322818	/	01.03.2019	NULL
2	1	8442	0.853861	1	0.453684	/bazen/	01.03.2019	NULL
3	1	125204	1	1	0.307322	/clovek/	01.03.2019	NULL
4	1	159	0.984178	1	0.597484	/obdobi/	01.03.2019	NULL
5	1	42437	1	1	0.303862	/plavec/	01.03.2019	NULL
6	1	10268	1	1	0.30298	/prihlaska/	01.03.2019	NULL
7	1	899	1	1	0.193548	/skupina/	01.03.2019	NULL
8	1	561	1	1	0.228164	/termin/	01.03.2019	NULL
9	1	1570	0.794693	1	0.538854	/vykon_kategorie/	01.03.2019	NULL

Tabulka 16 – datové objekty první a druhé úrovně pro první reálná data RD1

Dále Obrázek 32 zobrazuje veškerá data na čtyřech úrovních (databáze – tabulka – řádek - buňka) a to formou pokrytých oblastí, stejně jako v části 10.2, ovšem takto odděleným způsobem. Interval je opět roven $\frac{1}{20}$. Z obrázku je patrné, že na nejvyšší úrovni (obsahující pouze jeden, a to hlavní objekt) se z již zmíněných důvodů jako jediné projeví snížená míra hierarchičnosti 0,8762 a i míra informace 0,3228, neboť větší data snáze obsáhnou redundance. Naopak míra strukturovanosti je téměř absolutní (0.9997) – na vysoké úrovni jsou data velmi podrobně rozčleněná a drobné podíly nestrukturovaných podobjektů se projeví velmi málo. Na druhé úrovni je míra informace obdobná, ovšem je patrné, že některé tabulky ji mají výrazně vyšší (a jsou tak pravděpodobně méně redundantní, např. `obdobi` nebo `vykon_kategorie`) než jiné. To může opět souviset i s jejich menší velikostí a u některých tabulek je již patrná i nižší míra strukturovanosti. Na třetí úrovni již míra informace výrazně stoupá vzhledem k menší velikosti datových objektů a nižších hodnot blízkých 0,5 dosahuje jen ojediněle. Naopak míra strukturovanosti inklinuje ke krajním (zejména vyšším) hodnotám – zde se projeví řádky obsahující zcela strukturovaná data nebo naopak nestrukturované texty. Na čtvrté úrovni pak míra informace je velice vysoká a jen ojediněle ji snižuje drobný podíl buněk s nestrukturovanými daty, zatímco míra strukturovanosti už dosahuje výhradně krajních, binárních hodnot 0 a 1.



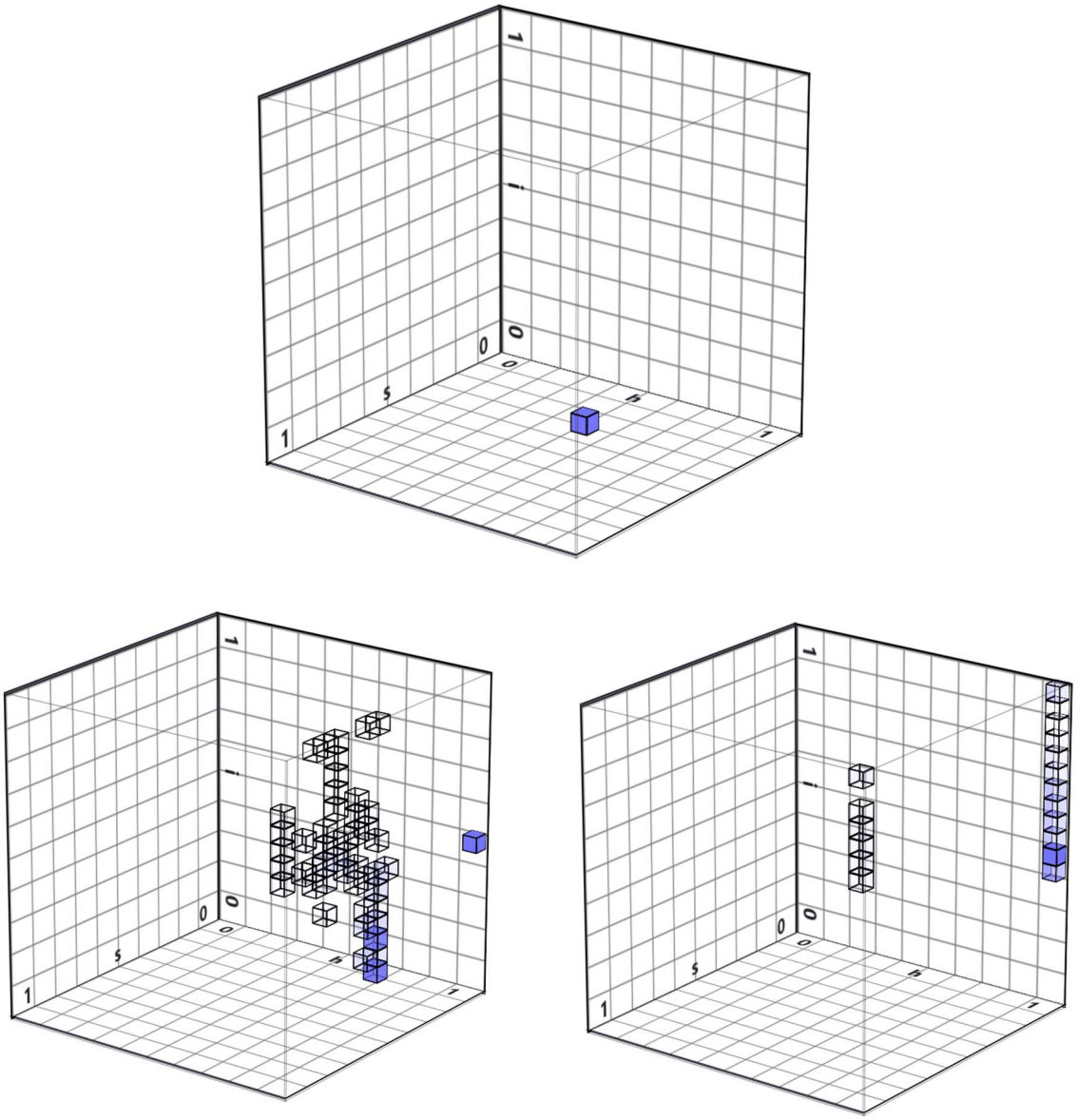
Obrázek 32 – vizualizace výsledků sestavení metadatového skladu pro reálná data RD1. Vlevo nahoře – 1. úroveň, vpravo nahoře – 2. úroveň, vlevo dole – 3. úroveň, vpravo dole – 4. úroveň

object_id	parent_object_id	size	structuredness	hierarchicality	information_amount	location_specification	existence_from	existence_to
1	NULL	126241	0.909079	1	0.306374	/	01.03.2019	NULL
2	1	126241	0.909079	1	0.306374	/text/	01.03.2019	NULL
3	2	742	0.77346	1	0.632075	/text/1/	01.03.2019	NULL
10	2	436	0.534325	1	0.644495	/text/13/	01.03.2019	NULL
17	2	1124	0.534535	1	0.356762	/text/41/	01.03.2019	NULL
24	2	2522	0.53376	1	0.500396	/text/46/	01.03.2019	NULL
31	2	817	0.829913	1	0.602203	/text/2/	01.03.2019	NULL
38	2	264	0.665504	1	0.598485	/text/43/	01.03.2019	NULL
45	2	83	0.738424	1	0.879518	/text/19/	01.03.2019	NULL

52	2	186	0.74208	1	0.698925	/text/75/	01.03.2019	NULL
59	2	125	0.768448	1	0.68	/text/21/	01.03.2019	NULL
66	2	136	0.74827	1	0.698529	/text/20/	01.03.2019	NULL
73	2	175	0.6128	1	0.662857	/text/24/	01.03.2019	NULL
80	2	199	0.69425	1	0.658291	/text/22/	01.03.2019	NULL
87	2	170	0.733875	1	0.611765	/text/23/	01.03.2019	NULL
94	2	146	0.720351	1	0.753425	/text/44/	01.03.2019	NULL
101	2	119	0.719723	1	0.764706	/text/18/	01.03.2019	NULL
108	2	35191	0.026857	1	0.494246	/text/47/	01.03.2019	NULL
115	2	695	0.754371	1	0.494964	/text/52/	01.03.2019	NULL
122	2	388	0.976332	1	0.662371	/text/72/	01.03.2019	NULL
129	2	943	0.7309	1	0.625663	/text/73/	01.03.2019	NULL
136	2	202	0.619645	1	0.752475	/text/50/	01.03.2019	NULL
143	2	940	0.713752	1	0.634043	/text/8/	01.03.2019	NULL
150	2	537	0.868634	1	0.700186	/text/5/	01.03.2019	NULL
157	2	593	0.636191	1	0.558179	/text/61/	01.03.2019	NULL
164	2	339	0.967195	1	0.690265	/text/3/	01.03.2019	NULL
171	2	89	0.842192	1	1	/text/74/	01.03.2019	NULL
178	2	2222	0.457339	1	0.538254	/text/48/	01.03.2019	NULL
185	2	327	0.64843	1	0.553517	/text/67/	01.03.2019	NULL
192	2	1575	0.521798	1	0.348571	/text/40/	01.03.2019	NULL
199	2	1941	0.512797	1	0.318908	/text/34/	01.03.2019	NULL
206	2	1646	0.54403	1	0.318955	/text/26/	01.03.2019	NULL
213	2	1230	0.534868	1	0.364228	/text/28/	01.03.2019	NULL
220	2	1845	0.518621	1	0.337127	/text/80/	01.03.2019	NULL
227	2	1675	0.529572	1	0.335522	/text/64/	01.03.2019	NULL
234	2	1016	0.579604	1	0.405512	/text/66/	01.03.2019	NULL
241	2	1226	0.570064	1	0.367863	/text/65/	01.03.2019	NULL
248	2	2063	0.525202	1	0.312651	/text/30/	01.03.2019	NULL
255	2	2307	0.514134	1	0.308626	/text/17/	01.03.2019	NULL
262	2	403	0.602707	1	0.51861	/text/16/	01.03.2019	NULL
269	2	2247	0.52221	1	0.319982	/text/29/	01.03.2019	NULL
276	2	1964	0.519047	1	0.327393	/text/54/	01.03.2019	NULL
283	2	1645	0.530212	1	0.346505	/text/42/	01.03.2019	NULL
290	2	1101	0.558884	1	0.372389	/text/27/	01.03.2019	NULL
297	2	3142	0.516259	1	0.270528	/text/58/	01.03.2019	NULL
304	2	2064	0.528326	1	0.315407	/text/59/	01.03.2019	NULL
311	2	781	0.543336	1	0.426376	/text/63/	01.03.2019	NULL
318	2	356	0.575559	1	0.505618	/text/49/	01.03.2019	NULL
325	2	1134	0.53424	1	0.362434	/text/39/	01.03.2019	NULL
332	2	407	0.557697	1	0.702703	/text/11/	01.03.2019	NULL
339	2	216	0.669753	1	0.638889	/text/25/	01.03.2019	NULL
346	2	1426	0.667259	1	0.598878	/text/12/	01.03.2019	NULL
353	2	18305	0.501037	1	0.226113	/text/45/	01.03.2019	NULL
360	2	78	0.555556	1	0.987179	/text/69/	01.03.2019	NULL
367	2	3410	0.501775	1	0.457478	/text/35/	01.03.2019	NULL
374	2	1106	0.764142	1	0.56962	/text/14/	01.03.2019	NULL
381	2	7427	0.529314	1	0.246533	/text/10/	01.03.2019	NULL
388	2	235	0.689108	1	0.642553	/text/77/	01.03.2019	NULL
395	2	102	0.719723	1	0.843137	/text/70/	01.03.2019	NULL
402	2	1187	0.541316	1	0.383319	/text/31/	01.03.2019	NULL
409	2	1232	0.552216	1	0.375812	/text/33/	01.03.2019	NULL
416	2	201	0.718547	1	0.80597	/text/9/	01.03.2019	NULL
423	2	260	0.987559	1	0.746154	/text/38/	01.03.2019	NULL
430	2	436	0.878788	1	0.642202	/text/76/	01.03.2019	NULL
437	2	722	0.78537	1	0.620499	/text/6/	01.03.2019	NULL
444	2	419	0.569688	1	0.670644	/text/4/	01.03.2019	NULL
451	2	171	0.72029	1	0.883041	/text/7/	01.03.2019	NULL
458	2	241	0.744305	1	0.672199	/text/32/	01.03.2019	NULL
465	2	368	0.982928	1	0.600543	/text/56/	01.03.2019	NULL

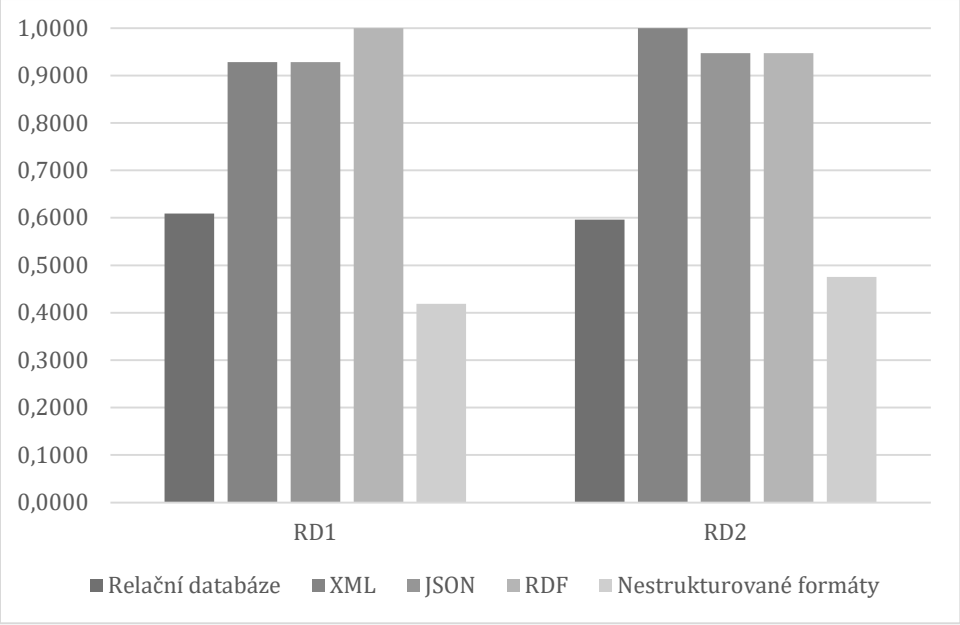
472	2	185	0.75214	1	0.708108	/text/55/	01.03.2019	NULL
479	2	56	0.831314	1	1	/text/53/	01.03.2019	NULL
486	2	66	0.757117	1	0.954545	/text/68/	01.03.2019	NULL
493	2	407	0.826138	1	0.572482	/text/78/	01.03.2019	NULL
500	2	874	0.543693	1	0.313501	/text/81/	01.03.2019	NULL
507	2	68	0.731618	1	0.985294	/text/51/	01.03.2019	NULL
514	2	122	0.622548	1	0.729508	/text/79/	01.03.2019	NULL
521	2	257	0.978546	1	0.81323	/text/37/	01.03.2019	NULL
528	2	844	0.561903	1	0.432464	/text/57/	01.03.2019	NULL
535	2	3882	0.562471	1	0.299073	/text/60/	01.03.2019	NULL
542	2	147	0.731778	1	0.945578	/text/62/	01.03.2019	NULL
549	2	257	0.98757	1	0.793774	/text/36/	01.03.2019	NULL
556	2	56	0.514987	1	1	/text/71/	01.03.2019	NULL

Tabulka 17 - datové objekty první, druhé a třetí úrovně pro druhá reálná data RD2



Obrázek 33 – vizualizace výsledků sestavení metadatového skladu pro druhá reálná data RD2. Nahoře – 1. úroveň, vlevo dole – 2. úroveň, vpravo dole – 3. úroveň

Obrázek 33 – vizualizace výsledků sestavení metadatového skladu pro druhá reálná data RD2. Nahoře – 1. úroveň, vlevo dole – 2. úroveň, vpravo dole – 3. úroveň opět znázorňuje jednotlivé úrovně metadatového skladu pomocí pokrytých oblastí krychlí o hraně délky $\frac{1}{20}$, ovšem 2. úroveň je vynechána; databáze obsahuje pouze jednu tabulku a metadatové objekty pro celou databázi a pro tuto tabulku mají shodné hodnoty metrik. Míra hierarchičnosti je, jak již bylo uvedeno a jak je na obrázcích vidět, v tomto případě vždy rovna jedné. První úroveň opět odpovídá pouze hlavnímu datovému objektu s velmi vysokou mírou strukturovanosti 0,9091, ovšem nižší mírou informace 0,3064. Tato hodnota indikuje značnou vzájemnou i vnitřní redundanci položek odpovídajících jednotlivým textům – v řadě případů sloupce `text_cz` a `text_en` mají shodný obsah. Tato jejich vzájemná redundance se projeví nejen na nejvyšší úrovni, ale na všech úrovních metadatového skladu kromě čtvrté, která odpovídá jednotlivým buňkám. Druhá úroveň již ukazuje značný podíl nestrukturovaných dat (způsobený jedním, ovšem velmi velkým řádkem 108) a jistý, byť nízký podíl dat o vyšší míře strukturovanosti i informace, tvořený zejména řádky obsahujícími velmi krátké texty. Třetí úroveň již stejně jako u prvních reálných dat RD1 rozlišuje míru strukturovanosti jako binární hodnotu s malým podílem hodnoty 1 (jde zejména o sloupce `id_text` a `ident`) a výrazně větším podílem hodnoty 0, kde míra informace dosahuje spíše středních hodnot.



Obrázek 34 – kompatibilita prvních reálných dat RD1 a druhých reálných dat RD2 s obecnými typy datových struktur

Typ datové struktury	RD1	RD2
Relační databáze	0,6090	0,5961
XML	0,9286	1,0000
JSON	0,9286	0,9475
RDF	0,9998	0,9475
Nestrukturované formáty	0,4184	0,4751

Tabulka 18 – kompatibilita hlavního objektu obou reálných dat a typů datových struktur

Tabulka 18 a Obrázek 34 uvádí jednotlivé míry kompatibility pro obě reálná data zjištěné stejným způsobem výpočtu, jako v případě dat demonstračních. Ačkoli v obou případech byla zkoumána faktická RDBS, kompatibilita s tímto typem datové struktury byla poměrně nízká, evidentně z důvodu nižších hodnot míry informace a tím vyššího podílu redundance.

Obecně jsou hodnoty kompatibility navzdory odlišným datům i výsledkům hodnot metrik pro oba hlavní datové objekty obou reálných dat podobné. Samotný hlavní objekt ovšem zcela nevypovídá o stavu svých datových podobjektů. Jak je vidět v tabulkách obou metadatových skladů, míra strukturovanosti na vyšší úrovni datového objektu při vysoké rozložitelnosti dat přirozeně musí být výrazně větší než u podobjektů. Míra informace naopak na vyšší úrovni klesá, protože větší velikost dat umožňuje větší výskyt redundance i opakujících se hodnot.

10.3.4 Dlouhodobé monitorování dat

Tato sekce demonstruje možnost využití metrik k dlouhodobému monitorování a s tím související automatizaci procesu výpočtu metrik v rámci metadatového skladu při opakovaném vícenásobném provedení v čase. Aplikovaný postup je zcela shodný s předchozími postupy v sekcích 10.2.1, 10.3.1, 10.3.2 a 10.3.3, které se také zaměřují na RDBS.

Jsou použity tři podmnožiny reálných dat uvedené na konci sekce 10.1.2., která se zabývá jejich výběrem. Vybrané datové podmnožiny se na základě vazby z tabulky `obdobi` vztahují postupně k rokům 2014, 2017 a 2020. Záměrem je ověřit, že postup je aplikovatelný i na dlouhodobější monitorování datové základny.

Pro hlavní datový objekt bylo dosaženo výsledků, které uvádí Tabulka 19. Dále Tabulka 20 uvádí výsledky pro podobjekty hlavního datového objektu, které reprezentují hodnoty pro jednotlivé tabulky RDBS.

míra strukturovanosti *s*	2014	0,9998
	2017	0,9989
	2020	0,9987
míra hierarchičnosti *h*	2014	0,1864
	2017	0,1893
	2020	0,2871
míra informace *i*	2014	0,2994
	2017	0,3465
	2020	0,3576

Tabulka 19 – hodnoty metrik hlavního datového objektu ve třech časových obdobích

		bazen	clovek	obdobi	plavec	prihlaska	skupina	termin	vykon_kategorie
s	**2014**	0,9136	1,0000	0,9642	1,0000	0,9996	1,0000	1,0000	0,7297
	2017	0,7760	1,0000	0,9947	1,0000	0,9994	1,0000	1,0000	0,7947
	2020	0,8782	1,0000	0,9947	1,0000	0,9993	1,0000	1,0000	0,7947
h	**2014**	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	2017	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	2020	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
i	**2014**	0,5289	0,3147	0,5135	0,1720	0,3415	0,1768	0,1258	0,5771
	2017	0,4783	0,3826	0,3676	0,2707	0,3608	0,1869	0,1627	0,5389
	2020	0,4577	0,3915	0,3606	0,3195	0,3505	0,2159	0,1777	0,5389

Tabulka 20 – hodnoty metrik datových objektů odpovídajících celým jednotlivým tabulkám RDBS daných reálných dat ve třech časových obdobích

Při výpočtech je i v tomto případě pracováno s velikostí dat coby prostou délkou textového řetězce, představujícího uspořádaná exportovaná data z databáze odpovídající konkrétnímu datovému objektu.

Z hlediska **míry strukturovanosti** se na úrovni celé databáze projevuje napříč časem zcela minimální změna. Při vyšší velikosti dat (délka exportovaného řetězce je 83166, 90725 a 94495 pro tato 3 vybraná období) je vyšší i rozložitelnost objektu a hodnota míry strukturovanosti je tedy blízká jedné. Totéž platí i pro některé dílčí tabulky o vyšší velikosti. Naopak u tabulek `bazen` a `vykon_kategorie` je hodnota míry strukturovanosti nižší zejména kvůli vysokému podílu nestrukturovaných textů (popisy sportovišť a výkonnostních kategorií účastníků). V případě tabulky `bazen` je navíc tato hodnota značně proměnlivá, kdy dochází k výraznému poklesu a následně opět vzestupu. Důvodem zřejmě je, že výběr provozovaných sportovišť se mění, jejich popisy jsou upravovány z marketingových důvodů a některá byla v roce 2020 v důsledku pandemie zcela uzavřena. Naopak tabulka `vykon_kategorie` obsahující popisy výkonnostních kategorií se evidentně téměř nemění.

Naopak z hlediska **míry hierarchičnosti** se jiná hodnota než 1,0000 objeví až na úrovni hlavního datového objektu (zde představujícího celou databázi). Jedna tabulka vzhledem ke schématu databáze (které zobrazuje Obrázek 25) nemůže sama o sobě porušit definici hierarchičnosti (viz Vzorec 1). Konkrétně je nižší míra hierarchičnosti vyvolána přítomností tabulky `prihlaska`, která obsahuje cizí klíče odkazující na tabulky `plavec` a `skupina`. Z časového hlediska je patrný nárůst míry hierarchičnosti především mezi roky 2017 a 2020, zřejmě z toho důvodu, že klesá velikost dat tabulky `prihlaska`, přičemž rozpis termínů se výrazně nemění a počet registrovaných osob je úměrný počtu přihlášek.

Z hlediska **míry informace** byly u hlavního datového objektu vypočteny poměrně nízké hodnoty 0,2994, 0,3465 a 0,3576 pro jednotlivé roky. To bylo způsobeno vysokým podílem velikosti tabulek, jejichž hodnota míry informace vyšla také nízká. U dílčích tabulek nejvyšších hodnot dosahují nejméně strukturované tabulky `bazen` a `vykon_kategorie` (viz míra strukturovanosti), které především pro nízkou velikost dat a vysoký podíl nestrukturovaných textů mají nejnižší četnost výskytu opakovaných fragmentů textu. Naopak nejnižších hodnot dosáhly tabulky `termin` a `skupina`, a to z důvodu svého číselníkového charakteru. Tyto tabulky také obsahují řadu opakujících se hodnot (termíny a skupiny v nich jsou v každém pololetí vypisovány stejně nebo velice

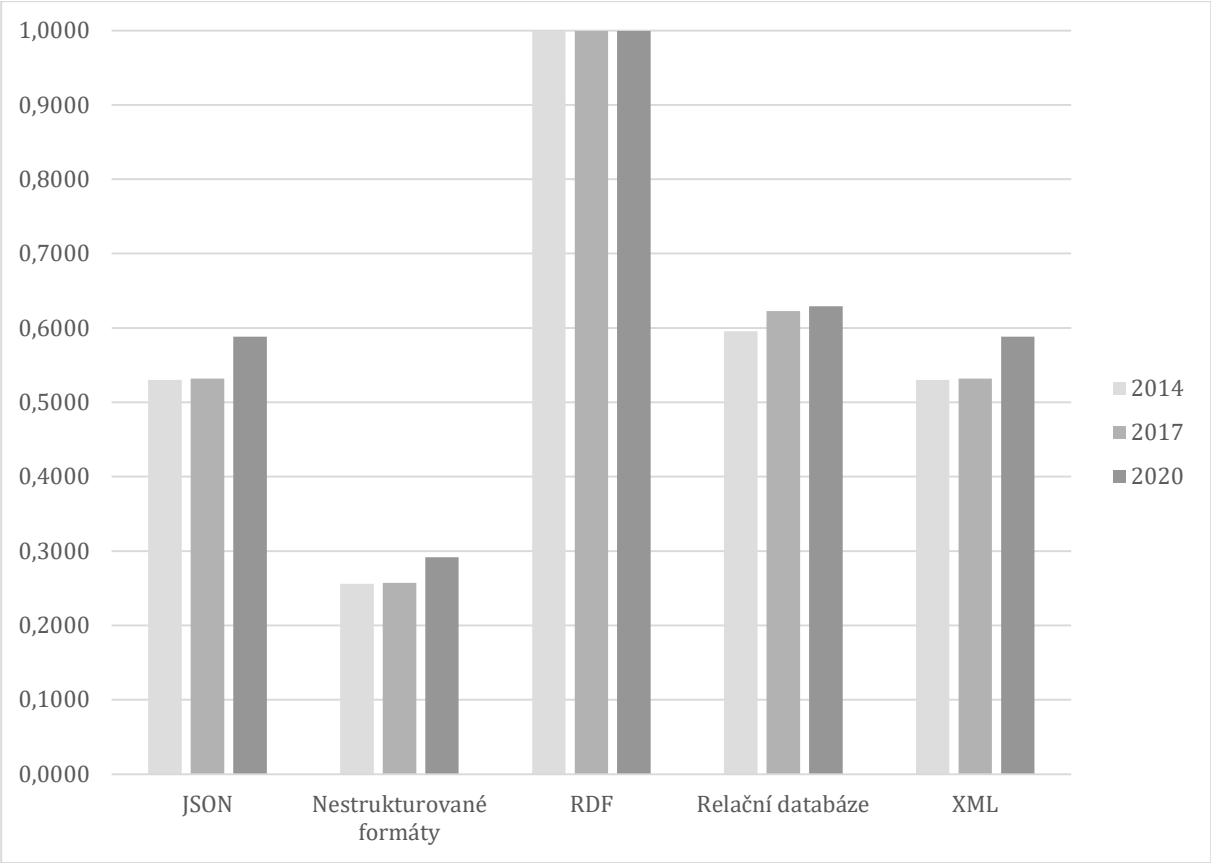
podobně). Nejedná se tedy přímo o výskyt redundance, nicméně na výpočet dle AIT to má výrazný vliv. Nejvýraznější pokles míry informace je zaznamenán u tabulky `obdobi`, která s každým pololetím získává další řádek s velmi podobným obsahem, což se také výrazně projeví na výpočtu dle AIT. Naopak nárůst je výrazný u tabulky `plavec`, která kromě jednoho primárního a cizího klíče obsahuje unikátní kód dotyčného účastníka, který pochopitelně není komprimovatelný. Pravděpodobně z podobných důvodů mírně narůstá hodnota i v tabulce `clovek`, obsahující z většiny jména, příjmení a e-maily.

Vstup experta je stejný, jak bylo stanoveno na začátku části 10.3. Tento expertův vstup nicméně je v případě periodické aplikace napříč časem potřebný pouze jednorázově. Procesy plnění metadatového skladu a výpočtu metrik se pak pro jednotlivé roky ukázaly jako automatizovatelné; bylo možné je bez úprav aplikovat na všechna tři časová období. Podmínkou byla neměnnost databázového schématu.

Je nutné upozornit, že vývoj datové základny v čase nelze posuzovat coby datovou transformaci tak, jak ji pojímá kapitola 7. Nejedná se totiž o změnu způsobu reprezentace jedné informace různými typy datových struktur, ale o opačný typ procesu, kdy stále stejným typem datové struktury jsou reprezentovány odlišné informace.

	2014	**2017**	**2020**
JSON	0,5303	0,5320	0,5884
Nestrukturované formáty	0,2558	0,2573	0,2916
RDF	0,9999	0,9994	0,9992
Relační databáze	0,5955	0,6227	0,6291
XML	0,5303	0,5320	0,5884

Tabulka 21 – míra kompatibility hlavního datového objektu s typy datových struktur ve třech analyzovaných rocích



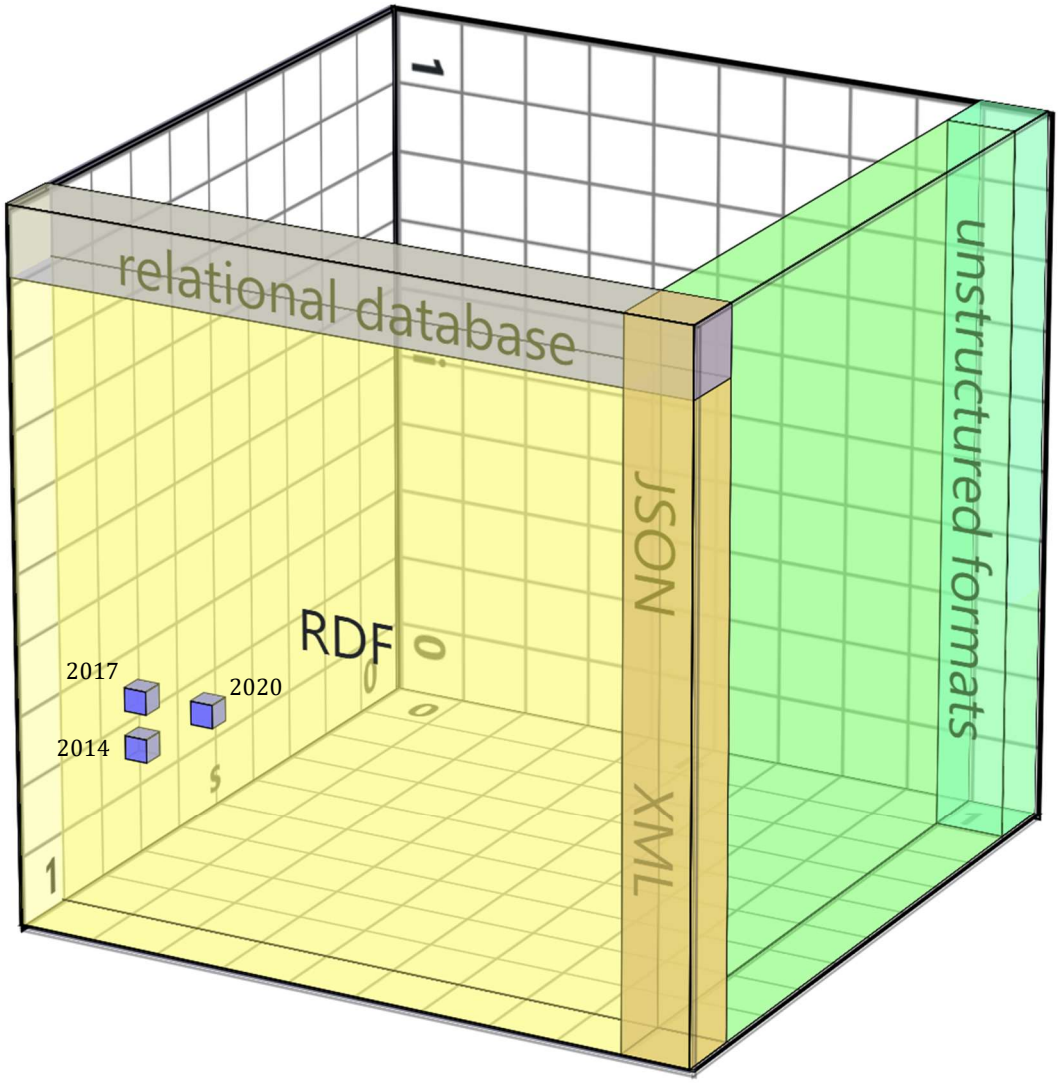
Obrázek 35 – graf hodnot míry kompatibility, které uvádí Tabulka 21

Tabulka 21 uvádí hodnoty **míry kompatibility** hlavních datových objektů pro roky a typy datových struktur. Tyto hodnoty dále vizualizuje Obrázek 35. Nejzajímavějším zjištěním je, že data jsou nejlépe reprezentovatelná formátem RDF, a to zřejmě v důsledku nepříliš vysokých hodnot míry informace. Jak bylo diskutováno již v části 6.3, technologie Linked data výskyt redundance umožňuje a vzhledem k jejímu decentralizovanému pojetí je často nutností.

Míra kompatibility s RDBS se ukázala jako nepříliš vysoká, byť s mírným růstovým trendem. Příčinou je opět častá nižší hodnota míry informace, na druhou stranu však na tuto hodnotu míry kompatibility měla pozitivní vliv vysoká hodnota míry strukturovanosti. Snížená míra hierarchičnosti pak měla negativní vliv na kompatibilitu s formáty XML a JSON. S nárůstem míry informace i hierarchičnosti všechny tyto tří hodnoty míry kompatibility (RDBS, XML i JSON) narůstaly.

Celkově lze z naměřených hodnot odvodit, že analyzovaná databáze může obsahovat řadu redundantních či jiným způsobem neoptimálně ukládaných hodnot a dlouhodobé srov-

nání ukazuje jen na velmi malé zlepšení. Její schéma by proto mělo být zrevidováno z hlediska normálních forem. Jako nejvíce problematická se jeví tabulka `termin`, jejíž struktura vyžaduje značné opakování shodných položek a patrně by mohla být optimalizována.



Obrázek 36 – vizualizace změn hodnot metrik hlavního datového objektu ve třech letech

Vizualizaci chronologického vývoje stavu hlavního datového objektu uvádí Obrázek 36. Vzhledem k vzájemné blízkosti dotyčných tří datových objektů (označených roky a propojených šipkami) mají pokryté oblasti trojrozměrného prostoru velikost hrany pouze $\frac{1}{30}$, aby nedocházelo k jejich „slepení“ či splynutí v jedinou krychli. Jsou vyznačeny dva typy datových struktur – RDBS (odpovídající datům, avšak s nižší hodnotou míry kompatibility) a RDF (s téměř maximální hodnotou míry kompatibility).

11 Závěr

11.1 Plnění cílů

Hlavním cílem práce bylo vytvoření výpočetní metody, která dokáže komplexně analyzovat charakteristiky informačních a datových objektů. Metoda vytvořena byla a při aplikaci na demonstračních i reálných data se ukázala jako použitelná. Jako hlavní omezení se ukázala její závislost na expertovi a jeho znalostech specifických zkoumaných dat, která zvyšuje pracnost úměrně velikosti a komplikovanosti struktury těchto dat. Dalším omezením je vyšší výpočetní náročnost při aplikaci na velmi rozsáhlých datech.

První dílčí cíl (DC1) spočíval v tvorbě metriky určující, do jaké míry jsou data strukturovaná. Byl navržen postup výpočtu metriky a následně demonstrován na řadě jednoduchých příkladů, které ukázaly splnění jeho očekávaných vlastností. Totéž bylo ukázáno i při aplikaci na dalších vzorcích demonstračních a reálných dat.

Druhý dílčí cíl (DC2) spočíval v tvorbě metriky určující, do jaké míry data reprezentují unikátní informaci (sdělení) a nikoli duplicity. Pro jednoduchost a univerzálnost byl aplikován princip algoritmické míry informace, který ukázal, že u explicitně redundantních dat dokáže prostřednictvím výpočtu vracet výrazně nižší hodnoty.

Třetí dílčí cíl (DC3) spočíval v tvorbě metriky určující, do jaké míry data splňují definici hierarchičnosti a jejich struktura tak je stromová. Při stanovování postupu výpočtu metriky bylo nutné vyřešit problém mnohonásobných porušení definice hierarchičnosti tak, aby případné překrývající se datové objekty nesnižovaly celkovou míru hierarchičnosti více, než odpovídá podílu jejich velikosti. Bylo navrženo řešení a hodnota metriky ukázala schopnost poukázat na oblasti dat, u nichž porušení hierarchičnosti vzniká.

Čtvrtý dílčí cíl (DC4) spočíval v tvorbě metody uložení a vzájemného srovnání výsledných hodnot metrik mezi objekty navzájem, vůči typům datových struktur, a i na časové ose (datové transformace). Tato metoda, nazvaná metadatový sklad navržený coby RDBS se ukázal jako zcela funkční úložiště vypočtených metrik. Vizualizační metoda ukázala, že hodnoty metrik, kompatibilitu datových objektů a typů datových struktur a transformace dat je možné vizualizovat ve trojrozměrném prostoru.

11.2 Zhodnocení přínosů

Práce otevřela téma unifikovaného přístupu ke klasifikaci dat bez přímé závislosti na typu datové struktury, a to prostřednictvím trojice metrik. Bylo ukázáno, že tyto metriky svými zcela rozdílnými optimálními hodnotami navzájem dělí jedny z nejčastějších typů datových struktur, a tedy umožňují klasifikovat i data s nimi do různé míry kompatibilní. Výpočet hodnot pro konkrétní data se podařilo formalizovat a demonstrovat na ukázkových datech. Bylo vytvořeno možné řešení způsobu ohodnocení informační a datové kompatibility. Bylo ukázáno, že kompatibilita odpovídá blízkosti geometrických útvarů v prostoru a že tuto blízkost lze vizualizovat.

Dílčí přínosy práce jsou tedy následující:

- formalizace výpočtu tří metrik na datovém objektu,
- popis typů datových struktur prostřednictvím návrhu optimálních intervalů hodnot metrik, jimž se hodnoty dat odpovídajících typu datové struktury v optimálním případě co nejvíce blíží,
- koncept rozkládání dat do stromu datových objektů od hlavního objektu až po elementární objekty,
- analýza možných i reálně přípustných transformačních směrů na základě kombinací změn hodnot metrik, znázornění jejich geometrické podstaty v prostoru a pomocí Vennova diagramu,
- koncept trojrozměrného metadatového prostoru a znázornění oblastí pokrytých datovými objekty,
- demonstrace analýzy dat prostřednictvím navržených metod.

11.3 Zhodnocení využitelnosti výsledků

Hodnoty změřených metrik lze využít v několika oblastech. První je **datové modelování** – nezávisle na aktuálním typu použité datové struktury je možné pomocí míry kompatibility prozkoumat blízkost vybraných předpokládaných dat k různým dalším typům datových struktur a vybrat nejvhodnější. Právě tak lze výpočet využít jako zpětnou vazbu v pozdějším čase, kdy datové modelování bylo dokončeno, datové struktury vybudovány a data již existují.

Navržená metoda umožňuje při delší kontinuální aplikaci na měnící se datové základně i **dlouhodobé monitorování** datové základny. Dokáže snadno upozornit na nárůst prosté velikosti dat a tu porovnat s algoritmickou mírou informace, čímž odhalí redundanci. Právě tak lze porovnat velikost dat s mírou strukturovanosti a odhalit nárůst nestrukturovaných dat, jejichž nízká míra strukturovanosti nemusí být žádoucí. Zjištěný nárůst redundance na základě algoritmické míry informace může být srovnán s nárůstem míry hierarchičnosti, což může indikovat nesprávně zvolený typ datové struktury, která nepodporuje normalizaci či vynucuje hierarchické uložení.

Navržená metoda umožňuje vyhledávat specifické podmnožiny dat vhodné k jistým účelům a určené k transformaci. Vyšší míra kompatibility a tím i nižší předpokládaná délka transformace může indikovat **menší náročnost převodu,** a tedy i nižší náklady (především z hlediska pracnosti experta) na provedení. Takto lze ohodnotit například prezentaci dat z relační databáze na webu, právě tak jako plnění relační databáze daty z webu, popřípadě z rozhraní (API) přenášejícího data ve formátu XML nebo JSON.

11.4 Náměty pro navazující a související výzkum

Na práci lze navázat plnou i částečnou automatizací řady provedených postupů. Dále navazující výzkum může problematiku rozšiřovat, a to o další zkoumané typy datových struktur, které mohou být zároveň inspirací i k zobecnění tří navržených metrik nebo návrhu dalších metrik. Mnoho prostoru existuje i v oblasti vizualizace stavu metadatových skladů, případně jejich sledování v reálném čase.

První oblastí umožňující automatizaci je proces samotného **budování metadatového skladu**. To je do jisté míry **automatizovatelné** pro nejtypičtější, tedy v práci uvedené datové struktury. Omezením je ovšem narážení na atypické datové konstrukty, jakými je například ukládání dokumentů XML a JSON do relační databáze, což řada relačních databázových systémů (například PostgreSQL) podporuje. Nejde ovšem o součást samotného konceptu SQL databází a standardu SQL a naopak se jedná o porušení normálních forem. Nelze ovšem vyloučit vývoj nástroje, který budování metadatového skladu přinejmenším výrazně usnadní pro běžné typy datových struktur a pracující expert se postará pouze o takovéto problémy pomocí algoritmů neřešitelné.

Druhou oblastí umožňující automatizaci je **určení optimálního směru datové transformace**. Například pomocí výpočtu míry hierarchičnosti může být transformace nasměrována simulací možných výsledků při vzniku minimální redundance a maximálního zachování míry strukturovanosti. Optimalizovat je možné i datové modelování nové struktury pro již existující data.

Dále je možný navazující teoretický výzkum celých tříd metrik, které by splňovaly v práci stanovené požadavky na určování míry informace, strukturovanosti a hierarchičnosti.

12 Seznamy

12.1 Seznam obrázků

12.2 Seznam tabulek

12.3 Seznam kódů a algoritmů

12.4 Seznam vzorců

12.5 Seznam zkratek

Pojem	**Zkratka**
Business Intelligence	BI
Algoritmická teorie informace	AIT
Multi-dimensional Management and Development of Information Systems	MMDIS
Small and medium enterprises	SME
Optical Character Recognition	OCR
Online Analytical Processing	OLAP
Online Transactional Processing	OLTP
Relational Database System	RDBS

13 Rejstřík pojmů

A

B

D

E

H

J

K

M

N

O

R

S

T

V

X

14 Použité informační zdroje a literatura

AASMAN, Jans, 2006. Allegro graph: RDF triple database. *Cidade: Oakland Franz Incorporated*.

ANDERSON, Kristi, 2019. *2019 Database Trends – SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use – High Scalability* - [online] [vid. 2019-10-24]. Dostupné z: http://highscalability.com/blog/2019/3/6/2019-database-trends-sql-vs-nosql-top-databases-single-vs-mu.html

ARENAS, Marcelo a Leonid LIBKIN, 2004. A normal form for XML documents. *ACM Transactions on Database Systems (TODS)*. **29**(1), 195–232.

ARENAS, Marcelo a Leonid LIBKIN, 2005. An information-theoretic approach to normal forms for relational and XML data. *Journal of the ACM (JACM)*. **52**(2), 246–283.

BARTMANN, Dieter, Freimut BODENDORF, Elmar J. SINZ a Otto K. FERSTL, 2011. *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. B.m.: University of Bamberg Press. ISBN 978-3-86309-010-4.

BASCI, Dilek a Sanjay MISRA, 2011. Entropy as a Measure of Quality of XML Schema Document. *The International Arab Journal of Information Technology*. **8**(1), 75–83.

BEACH, Christopher S. a William R. SCHIEFELBEIN, 2014. Unstructured Data. *Journal of Accountancy*. **217**(1), 46–51. ISSN 00218448.

BECKER, Bob, 2003. Design Tip #46: Another Look At Degenerate Dimensions. *Kimball Group* [online] [vid. 2018-02-12]. Dostupné z: https://www.kimballgroup.com/2003/06/design-tip-46-another-look-at-degenerate-dimensions/

BEGG, Carolyn a Tom CAIRA, 2012. Exploring the SME Quandary: Data Governance in Practise in the Small to Medium-Sized Enterprise Sector. *Electronic Journal of Information Systems Evaluation*. **15**(1), 3–13. ISSN 15666379.

BELLINGER, Gene, Durval CASTRO a Anthony MILLS, 2004. Data, information, knowledge, and wisdom [online]. [vid. 2016-01-14]. Dostupné z: http://geoffreyanderson.net/capstone/export/37/trunk/research/ackoffDiscussion.pdf

BOEHM, B. W. ET AL., 1973. Characteristics of software quality. In: *TRW-SS-73-09*.

BUCHALCEVOVÁ, Alena, nedatováno. Metodický rámec budování IS/ICT [online]. **2004**(3). Dostupné z: https://www.researchgate.net/publication/267402361_Metodicky_ramec_budovani_ISICT

BUSINESS PERFORMANCE PTY LTD, nedatováno. *Hard Measures versus Soft Measures* [online] [vid. 2017-12-03]. Dostupné z: http://www.businessperform.com/workplace-training/hard-measures-versus-soft-measures.html

BUSINESSDICTIONARY, nedatováno. What is psuedo bill of material? definition and meaning. *BusinessDictionary.com* [online] [vid. 2019-08-07]. Dostupné z: http://www.businessdictionary.com/definition/psuedo-bill-of-material.html

CARDELLI, Luca, 2001. Describing semistructured data. *ACM SIGMOD Record*. **30**(4), 80–85.

CERRA, Daniele, Alexandre MALLET, Lionel GUEGUEN a Mihai DATCU, 2010. Algorithmic Information Theory-Based Analysis of Earth Observation Images: An Assessment. *IEEE Geoscience and Remote Sensing Letters* [online]. **7**(1), 8–12. ISSN 1545-598X. Dostupné z: doi:10.1109/LGRS.2009.2020349

CLARK, Patrick G., Jerzy W. GRZYMALA-BUSSE a Wojciech RZASA, 2015. Consistency of incomplete data. *Information Sciences* [online]. **322**, 197–222. ISSN 00200255. Dostupné z: doi:10.1016/j.ins.2015.06.011

CODD, E. F., 1990. *The relational model for database management: version 2*. Reading, Mass: Addison-Wesley. ISBN 978-0-201-14192-4.

CUPOLI, Patricia, Susan EARLEY a Deborah HENDERSON, 2012. DAMA-DMBOK2 Framework. 27.

DAMA INTERNATIONAL, 2017. *DAMA-DMBOK: Data Management Body of Knowledge: 2nd Edition*. Second edition. Basking Ridge, New Jersey: Technics Publications. ISBN 978-1-63462-234-9.

DANIEL VODŇANSKÝ, 2020. 3D data metrics visualizer. *3D data metrics visualizer* [online] [vid. 2021-02-21]. Dostupné z: https://danielvodnansky.github.io/3d-data-histogram/

DATA GOVERNANCE INSTITUTE, nedatováno. The DGI Data Governance Framework. *The DGI* [online]. [vid. 2016-07-09]. Dostupné z: http://www.datagovernance.com/wp-content/uploads/2014/11/dgi_framework.pdf

DE MAURO, Andrea, Marco GRECO a Michele GRIMALDI, 2016. A formal definition of Big Data based on its essential features. *Library Review* [online]. **65**(3), 122–135. ISSN 0024-2535. Dostupné z: doi:10.1108/LR-06-2015-0061

DORAN, Paul, Valentina TAMMA, Ignazio PALMISANO, Terry R PAYNE a Luigi IANNONE, 2008. Evaluating ontology modules using an entropy inspired metric. In: *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*. B.m.: IEEE, s. 918–922. ISBN 0-7695-3496-1.

DUDÁŠ, Marek, Steffen LOHMANN, Vojtěch SVÁTEK a Dmitry PAVLOV, 2018. Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review* [online]. **33**. Dostupné z: doi:10.1017/S0269888918000073

EBERHART, Aaron, David CARRAL, Pascal HITZLER, Hilmar LAPP a Sebastian RUDOLPH, nedatováno. Seed Patterns for Modeling Trees. 20.

EXPLORE GROUP, 2019. The Most Popular Databases 2019 | Blog. *Explore Group* [online] [vid. 2019-10-24]. Dostupné z: https://www.explore-group.com/blog/the-most-popular-databases-2019/bp46/

FEUERLICHT, George, Vladimir KOVAR, David HARTMAN, Marek BERANEK a Pavel BORY, 2015. Measuring Complexity of Domain Standard Specifications Using XML Schema Entropy. In: *SOFSEM (Student Research Forum Papers/Posters)* [online]. s. 124–131 [vid. 2016-03-20]. Dostupné z: http://ceur-ws.org/Vol-1326/124-Feuerlicht.pdf

FLORESCU, Daniela, 2005. Managing Semi-Structured Data. *Queue* [online]. **3**(8), 18–24. ISSN 1542-7730. Dostupné z: doi:10.1145/1103822.1103832

FLORIDI, Luciano, 2013. Information Quality. *Philosophy & Technology; Dordrecht* [online]. **26**(1), 1–6. ISSN 22105433. Dostupné z: doi:http://dx.doi.org.zdroje.vse.cz:2048/10.1007/s13347-013-0101-3

FOOTE, Keith D., 2019. So You Want to be a Data Curator? *DATAVERSITY* [online]. [vid. 2019-06-16]. Dostupné z: https://www.dataversity.net/so-you-want-to-be-a-data-curator/

GALAR, M., H. BUSTINCE, J. FERNANDEZ, J. SANZ a G. BELIAKOV, 2010. Fuzzy Entropy from Weak Fuzzy Subsethood Measures. *Neural Network World*. **20**(1), 139–158. ISSN 12100552.

GANGEMI, Aldo, Valentina PRESUTTI, Diego REFORGIATO RECUPERO, Andrea Giovanni NUZZOLESE, Francesco DRAICCHIO a Misael MONGIOVÌ, 2017. Semantic web machine reading with FRED. *Semantic Web*. **8**(6), 873–893.

GARTNER, 2017. *Gartner IT Glossary > Big Data – From the Gartner IT Glossary: What is Big Data?* [online]. 18. červenec 2017. Dostupné z: https://web.archive.org/web/20170718161704/https://research.gartner.com/definition-whatis-big-data

GRÜNWALD, Peter D. a Paul MB VITÁNYI, 2008. Algorithmic information theory. *Handbook of the Philosophy of Information*. 281–320.

GUERRERO, Fabio G., 2012. Sobre la entropía del español escrito. *Revista Colombiana de Estadística*. **35**(3), 423–440. ISSN 2389-8976.

HALPIN, Terry, 2001. *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-672-2.

HEATH, Tom a Christian BIZER, 2011. Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology* [online]. **1**(1), 1–136. ISSN 2160-4711. Dostupné z: doi:10.2200/S00334ED1V01Y201102WBE001

HELLAND, Pat, 2017. XML and JSON Are Like Cardboard. *Communications of the ACM* [online]. **60**(12), 46–47. ISSN 00010782. Dostupné z: doi:10.1145/3132269

HOVEN, John van den, 2003. Data Architecture: Principles for Data. *Information Systems Management* [online]. **20**(3), 93–96. ISSN 1058-0530. Dostupné z: doi:10.1201/1078/43205.20.3.20030601/43078.11

HOVEN, John van den, 2004. Data Architecture Standards for the Effective Enterprise. *Information Systems Management* [online]. **21**(3), 61–64. ISSN 1058-0530. Dostupné z: doi:10.1201/1078/44432.21.3.20040601/82478.9

HRADIL, Jiří a Vilém SKLENÁK, 2017. Practical Implementation of 10 Rules for Writing REST APIs. *Journal of Systems Integration* [online]. **8**(1), 45-54–54. ISSN 1804-2724. Dostupné z: doi:10.20470/jsi.v8i1.290

HUTTER, Marcus, 2007. Algorithmic information theory. *Scholarpedia* [online]. **2**(3), 2519. ISSN 1941-6016. Dostupné z: doi:10.4249/scholarpedia.2519

CHAUDHURI, Surajit a Umeshwar DAYAL, 1997. An overview of data warehousing and OLAP technology. *ACM Sigmod record*. **26**(1), 65–74.

ISACA, 2019. *COBIT 2019 Governance and Management Objectives* [online] [vid. 2019-06-01]. Dostupné z: http://www.isaca.org/COBIT/Pages/COBIT-2019-Design-Guide.aspx

KANEHISA, Minoru, Susumu GOTO, Yoko SATO, Masayuki KAWASHIMA, Miho FURUMICHI a Mao TANABE, 2014. Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Research* [online]. **42**(D1), D199–D205. ISSN 0305-1048, 1362-4962. Dostupné z: doi:10.1093/nar/gkt1076

KENT, William, 1983. A simple guide to five normal forms in relational database theory. *Communications of the ACM*. **26**(2), 120–125.

KOLMOGOROV, Andrei N., 1963. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*. 369–376.

KOSKO, Bart, 1986. Fuzzy entropy and conditioning. *Information Sciences* [online]. **40**(2), 165–174. ISSN 0020-0255. Dostupné z: doi:10.1016/0020-0255(86)90006-X

KRISHNAMURTHY, Rajasekar, Jeffrey F. NAUGHTON, Jayavel SHANMUGASUNDARAM a Eugene SHEKITA, 2001. Dealing with (un) structuredness in XML Data and Queries Using Relational Databases. In: *DB seminar at Wise university* [online]. [vid. 2017-10-15]. Dostupné z: https://pdfs.semanticscholar.org/acb6/72e6feea4893192c74fc4cf3dcce31b3ad65.pdf

LAUE, Ralf a Jan MENDLING, 2010. Structuredness and its significance for correctness of process models. *Information Systems and e-Business Management* [online]. **8**(3), 287–307. ISSN 1617-9846, 1617-9854. Dostupné z: doi:10.1007/s10257-009-0120-x

LESLIE F. SIKOS, nedatováno. Knowledge Base = TBox + ABox. *Knowledge Base* [online] [vid. 2017-12-17]. Dostupné z: http://www.lesliesikos.com/knowledge-base/

LIEW, Anthony, 2007. Understanding Data, Information, Knowledge And Their Inter-Relationships. *Journal of Knowledge Management Practice*. **Vol. 7**.

LNĚNIČKA, Martin, 2015. AHP Model for the Big Data Analytics Platform Selection. *Acta Informatica Pragensia*. **4**(2), 108–121. ISSN 1805-4951.

LOH, Peter a Deepak SUBRAMANIAN, 2010. Fuzzy Classification Metrics for Scanner Assessment and Vulnerability Reporting. *IEEE Transactions on Information Forensics and Security*. **5**, 613–624.

MADISON, Michael, Mark BARNHILL, Cassie NAPIER a Joy GODIN, 2015. NoSQL Database Technologies. *Journal of International Technology & Information Management*. **24**(1), 1–13. ISSN 15435962.

MBI, nedatováno. *DIG0 : Dimenze - úvod, souhrnný přehled* [online] [vid. 2017-10-13]. Dostupné z: http://mbi.vse.cz/mbi/index.html#obj/DIMENGROUP-14

MEINSMA, Gjerrit, nedatováno. Data compression & Information theory [online]. **2014**. Dostupné z: https://www.yumpu.com/en/document/view/27882302/data-compression-information-theory

MICROSOFT, 2017. *XML Data (SQL Server)* [online] [vid. 2017-09-17]. Dostupné z: https://docs.microsoft.com/en-us/sql/relational-databases/xml/xml-data-sql-server

MORTON, John, ed., 2014. *Big data: opportunities and challenges*. Swindon: BCS, The Chartered Institute for IT.

MUSCA, Serban C., Rodolphe KAMIEJSKI, Armelle NUGIER, Alain MÉOT, Abdelatif ER-RAFIY a Markus BRAUER, 2011. Data with Hierarchical Structure: Impact of Intraclass Correlation and Sample Size on Type-I Error. *Frontiers in Psychology* [online]. **2** [vid. 2016-02-13]. ISSN 1664-1078. Dostupné z: doi:10.3389/fpsyg.2011.00074

NÄRMAN, Per, Hannes HOLM, Pontus JOHNSON, Johan KÖNIG, Moustafa CHENINE a Mathias EKSTEDT, 2011. Data accuracy assessment using enterprise architecture. *Enterprise Information Systems* [online]. **5**(1), 37–58. ISSN 1751-7575. Dostupné z: doi:10.1080/17517575.2010.507878

NEČASKÝ, Martin, 2006. Conceptual modeling for XML: A survey. *Databases, Texts*. 40.

NEČASKÝ, Martin, Jaroslav POKORNÝ, Karel RICHTA, Kamil TOMAN a Vojtěch TOMAN, 2008. *XML technologie: Principy a aplikace v praxi*. B.m.: Grada Publishing a.s. ISBN 978-80-247-6688-1.

NIEWERTH, Matthias a Thomas SCHWENTICK, 2018. Reasoning About XML Constraints Based on XML-to-Relational Mappings. *Theory of Computing Systems* [online]. **62**(8), 1826–1879. ISSN 14324350. Dostupné z: doi:10.1007/s00224-018-9846-5

ORACLE, nedatováno. *Oracle Technology Network > XML DB home* [online] [vid. 2017-09-17]. Dostupné z: http://www.oracle.com/technetwork/database/database-technologies/xmldb/overview/index.html

OREN, Eyal, Knud MÖLLER, Simon SCERRI, Siegfried HANDSCHUH a Michael SINTEK, 2006. What are semantic annotations. *Relatório técnico. DERI Galway*. **9**, 62.

PEJČOCH, David, 2014. CADAQUES: Metodika pro komplexní řízení kvality dat a informací. *Acta Informatica Pragensia* [online]. **3**(1), 44–56. ISSN 18054951, 18054951. Dostupné z: doi:10.18267/j.aip.35

POKORNÝ, Jaroslav, 2010. Databases in the 3rd Millennium: Trends and Research Directions. *Journal of Systems Integration*. **1**(1–2), 3–15. ISSN 1804-2724.

RAMEL, David, 2015. Relational Databases Still Reign in Enterprises, Survey Says. *Enterprise Systems* [online]. **2015** [vid. 2017-09-17]. Dostupné z: https://esj.com/articles/2015/04/23/database-survey.aspx

RAMEL, David, 2019. Database Trends Report: SQL Beats NoSQL, MySQL Most Popular -. *ADTmag* [online] [vid. 2019-10-24]. Dostupné z: https://adtmag.com/articles/2019/03/05/db-report.aspx

ROBERTS, Paige, 2010. Corralling Unstructured Data for Data Warehouses. *Business Intelligence Journal*. **15**(4), 50–55. ISSN 15472825.

RUELLAN, Hervé, 2012. XML Entropy Study. In: *Balisage: The Markup Conference* [online]. s. 7–10 [vid. 2016-03-19]. Dostupné z: http://www.balisage.net/Proceedings/vol8/print/Ruellan01/BalisageVol8-Ruellan01.html

SCALEGRID.IO, 2019. *2019 Database Trends - SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use* [online] [vid. 2019-08-04]. Dostupné z: https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/

SHAH, Dharmesh, 2013. *Measuring What Matters: How To Pick A Good Metric* [online] [vid. 2016-06-26]. Dostupné z: http://onstartups.com/tabid/3339/bid/96738/Measuring-What-Matters-How-To-Pick-A-Good-Metric.aspx

SHANNON, C. E., 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* [online]. **27**(3), 379–423. Dostupné z: doi:10.1002/j.1538-7305.1948.tb01338.x

SHEPELEV, I. E., 2011. Identification of the hierarchical data structure. *Pattern Recognition and Image Analysis* [online]. **21**(2), 211–214. ISSN 10546618. Dostupné z: doi:http://dx.doi.org.zdroje.vse.cz/10.1134/S1054661811020994

SOLID IT, 2019. DB-Engines Ranking. *DB-Engines* [online] [vid. 2019-10-24]. Dostupné z: https://db-engines.com/en/ranking

SVÁTEK, Vojtěch, 2002. Ontologie a WWW. In: *Sborník konference Datakon*. s. 27–55.

SVÁTEK, Vojtěch a Miroslav VACURA, 2007. Ontologické inženýrství. In: *Sborník konference Datakon 2007, 20.-23. října 2007 Brno, Česká republika* [online]. s. 60–91 [vid. 2016-01-16]. Dostupné z: http://www.researchgate.net/profile/Miroslav_Vacura/publication/228650805_Ontologick_inenrstv/links/0fcfd50644b30b399d000000.pdf

SVÁTEK, Vojtěch, Ondřej ZAMAZAL, Christian MEILICKE a Heiner STUCKENSCHMIDT, nedatováno. OntoFarm Project. *OntoFarm Project* [online] [vid. 2016-12-06]. Dostupné z: http://owl.vse.cz:8080/ontofarm/

ŠPERKOVÁ, Lucie, 2014. Unstructured Data Analysis from Facebook Banking Sites. *Acta Informatica Pragensia*. **2014**(2), 154–167. ISSN 1805-4951.

TANG, Ruiming, Huayu WU a Stéphane BRESSAN, 2011. Measuring XML Structured-ness with Entropy. In: Liwei WANG, Jingjue JIANG, Jiaheng LU, Liang HONG a Bin LIU, ed. *Web-Age Information Management* [online]. B.m.: Springer Berlin Heidelberg, Lecture Notes in Computer Science, 7142, s. 113–123 [vid. 2016-06-18]. ISBN 978-3-642-28634-6. Dostupné z: doi:10.1007/978-3-642-28635-3_10

THADANI, Kapil a Kathleen MCKEOWN, 2008. A Framework for Identifying Textual Redundancy. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)* [online]. Manchester, UK: Coling 2008 Organizing Committee, s. 873–880 [vid. 2019-04-08]. Dostupné z: https://www.aclweb.org/anthology/C08-1110

THAW, Tin Zar a Mie Mie KHIN, 2011. Entropy measure of XML schema document complexity metric. In: *Computer Research and Development (ICCRD), 2011 3rd International Conference on* [online]. B.m.: IEEE, s. 476–479 [vid. 2014-11-05]. Dostupné z: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5764178

THAW, Tin Zar a Mie Mie KHIN, 2013. Measuring Qualities of XML Schema Documents. *Journal of Software Engineering and Applications* [online]. **06**(09), 458–469. ISSN 1945-3116, 1945-3124. Dostupné z: doi:10.4236/jsea.2013.69056

THE OPEN GROUP, 2017. *ArchiMate® 3.0.1 Specification* [online] [vid. 2019-08-07]. Dostupné z: https://pubs.opengroup.org/architecture/archimate3-doc/chap08.html

THE OPEN GROUP, nedatováno. *TOGAF®, an Open Group standard | The Open Group* [online] [vid. 2016-01-17]. Dostupné z: http://www.opengroup.org/subjectareas/enterprise/togaf

TOBIN, Daniel R., 1996. *Transformational Learning: Renewing Your Company Through Knowledge and Skills*. 1 edition. New York: Wiley. ISBN 978-0-471-13289-9.

UČEŇ, Pavel, nedatováno. *Optimální dimenzování informatiky* [online] [vid. 2017-12-03]. Dostupné z: http://www.systemonline.cz/clanky/optimalni-dimenzovani-informatiky.htm

VANDENBUSSCHE, Pierre-Yves, Ghislain A. ATEMEZING, María POVEDA-VILLALÓN a Bernard VATANT, 2017. Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web. *Semantic Web*. **8**(3), 437–452.

VODŇANSKÝ, Daniel, 2016a. Entropy-based hierarchization of relational data structures. *Journal of Systems Integration* [online]. **7**(4), 25–34. Dostupné z: doi:10.20470/jsi.v7i4.275

VODŇANSKÝ, Daniel, 2016b. *Metriky integrovaných podnikových dat v kontextu SME | Česká společnost pro systémovou integraci* [online] [vid. 2016-11-13]. Dostupné z: http://www.cssi.cz/cssi/metriky-integrovanych-podnikovych-dat-v-kontextu-sme

VODŇANSKÝ, Daniel a Ondřej ZAMAZAL, 2016. Study on Graph Metrics over Linked Open Vocabularies and OntoFarm Collections. In: *International Conference on Knowledge Engineering and Semantic Web*: *7th International Conference, KESW 2016, Prague, Czech Republic, September 21-23, 2016, Proceedings* [online]. Prague: Springer, s. 1–2. Dostupné z: http://2016.kesw.ru/

VOŘÍŠEK, Jiří, 2008. *Principy a modely řízení podnikové informatiky*. Praha: Oeconomica. ISBN 978-80-245-1440-6.

VOŘÍŠEK JIŘÍ, nedatováno. *05MMDIS-Princip_multidimezionality_a_dimenze_reseni_IS.zip* [online] [vid. 2016-01-23]. Dostupné z: http://nb.vse.cz/~vorisek/FILES/4IT215_materialy_k_predmetu/05MMDIS-Princip_multidimezionality_a_dimenze_reseni_IS.zip

VOŘÍŠEK, Jiří, nedatováno. *MBI - Management Byznys Informatiky - home page* [online] [vid. 2016-08-01]. Dostupné z: http://mbi.vse.cz/

VOŘÍŠEK, Jiří a Jan POUR, 2012. *Management podnikové informatiky*. 1. vyd. Praha: Professional Publishing. ISBN 978-80-7431-102-4.

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE, nedatováno. *Studijní oddělení – Fakulta informatiky a statistiky – Vysoká škola ekonomická v Praze* [online] [vid. 2019-04-06]. Dostupné z: https://fis.vse.cz/fakulta/organy-fakulty/studijni-oddeleni/

WANG, Liwei, Jingjue JIANG, Jiaheng LU, Liang HONG a Bin LIU, 2012. *Web-Age Information Management: WAIM 2011 International Workshops: WGIM 2011, XMLDM 2011, SNA 2011, Wuhan, China, September 14-16, 2011, Revised Selected Papers*. B.m.: Springer. ISBN 978-3-642-28635-3.

WUWONGSE, Vilas, Kiyoshi AKAMA, Chutiporn ANUTARIYA a Ekawit NANTAJEEWARAWAT, 2003. A Data Model for XML Databases. *Journal of Intelligent Information Systems*. **20**(1), 63. ISSN 09259902.

ZACH, Ondřej, Bjørn Erik MUNKVOLD a Dag Håkon OLSEN, 2014. ERP system implementation in SMEs: exploring the influences of the SME context. *Enterprise Information Systems* [online]. **8**(2), 309–335. ISSN 17517575. Dostupné z: doi:10.1080/17517575.2012.702358

ZACHMAN, John P., 2009. The Zachman framework evolution. *EA Articles (page intentionally left blank)* [online]. [vid. 2016-03-18]. Dostupné z: http://www.cob.unt.edu/itds/faculty/becker/BCIS5520/Readings/The%20Zachman%20Framework%E2%84%A2%20Evolution.pdf