

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Dotazovací jazyk pro vztahy MBI objektů

Ondřej Batík

Vedoucí práce: Ing. Michal Valenta, Ph.D.

10. května 2015

Poděkování

V první řadě bych chtěl poděkovat vedoucímu mé práce za jeho vstřícný přístup a všechny rady, které mi v průběhu její tvorby poskytl. Poděkování patří také mé rodině a nejbližším, kteří mě podporovali během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 10. května 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Ondřej Batík. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Batík, Ondřej. *Dotazovací jazyk pro vztahy MBI objektů*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Cílem práce je analýza uživatelských požadavků pro dotazování se na vztahy mezi objekty informační báze MBI (Management Byznys Informatiky), návrh nového dotazovacího jazyka nad těmito objekty a návrh implementace.

Na základě analýzy, která se zaměřuje především na potřeby dotazování uživatelů a současný stav MBI, je definováno nové schéma datového úložiště. Nad ním je následně navržen dotazovací jazyk, vycházející také z výše zmiňované analýzy. Výsledný jazyk je dále porovnáván s jiným doménově specifickým dotazovacím jazykem. Realizovatelnost řešení je prokázána formou „proof-of-concept“. Práce mimo jiné stručně pojednává o MBI, grafové databázi Neo4j a bakalářské práci Bc. Vojtěcha Stránského, na kterou volně navazuje.

Výsledkem je tedy sepsaná analýza, komplexní popis nového schématu datového úložiště a návrh řešení dotazovacího jazyka spolu s implementačním návrhem. Uživatelé by měla být poskytnuta alternativní možnost pro získávání dat z MBI objektů.

Klíčová slova dotazovací jazyk, MBI objekty, schéma datového úložiště, analýza požadavků, grafová databáze Neo4j, grafické prvky

Abstract

The goal of this work (bachelor thesis) is an analysis of user requirements for querying on relationships between the objects of the information base MBI (Management of Business Informatics), the design of a new query language above these objects and the design of the implementation.

Based on the analysis, which is primarily focused on the needs for querying by users and the current status of MBI, the new data storage scheme is defined. Then above this scheme the query language is designed, outgoing also from the aforementioned analysis. The resultant language is compared with another domain-specific query language afterwards. The feasibility of solution is demonstrated by the proof-of-concept form. The work also briefly discusses about the MBI, the graph database Neo4j and the bachelor thesis of Bc. Vojtěch Stránský, which the work further develops.

The outcome is written analysis, the comprehensive description of the new data storage scheme and the design of the query language solution together with the implementation design. The alternative option for data mining from MBI objects should be provided for the users.

Keywords query language, MBI objects, data storage scheme, analysis of requirements, graph database Neo4j, graphical elements

Obsah

Úvod	1
Motivace	1
Struktura práce	1
1 MBI a související práce	3
1.1 MBI - Základní informace	3
1.2 Struktura modelu - objekty MBI	4
1.3 Vizualizace vztahů v informační bázi MBI	7
2 Analýza uživatelských potřeb	9
2.1 Uživatelé a jejich dotazy	9
2.2 Analýza současného stavu portálu	10
2.3 Co je dobré a co by šlo vylepšit?	15
2.4 Výsledek analýzy	17
3 Cílová technologie	19
3.1 Grafové databáze a Neo4j	19
3.2 Jazyk Cypher	23
4 Analýza starého schématu datového úložiště	25
4.1 Struktura uzlů	25
4.2 Struktura hran	26
4.3 Import dat do úložiště	26
5 Návrh nového schématu datového úložiště	29
5.1 Důvody vzniku nového schématu	29
5.2 Nové schéma datového úložiště	30
5.3 Schéma v praxi	34
6 Návrh dotazovacího jazyka	37

6.1	Průběh návrhu	37
6.2	Specificky orientované dotazování	40
6.3	Dotazování pomocí vazeb	42
6.4	Asistované procházení grafem	48
7	Porovnání jazyka s jiným	51
7.1	Grafický dotazovací jazyk GQL	51
7.2	Porovnání jazyků	55
8	Návrh implementace	57
8.1	Modely aplikace	57
8.2	Návrh koncepce a implementační prostředí	59
8.3	Proof-of-concept	61
	Závěr	65
	Literatura	67
A	Seznam použitých zkratk	69
B	Uživatelská příručka a další přílohy	71
B.1	Uživatelská příručka	71
B.2	Konfigurační soubory	71
B.3	Dotazy	71
C	Obsah příloženého CD	73

Seznam obrázků

1.1	Model informační báze MBI.	4
2.1	Postranní menu po přihlášení do portálu.	11
2.2	Výpis instancí objektu Úloha.	12
2.3	Souhrnný slide k instanci: „Řízení komunikace vedení s IT byznysem“.	13
2.4	Ukázka dotazování přes hierarchickou strukturu objektu Úloha.	14
2.5	Výpis výsledků vyhledávání řetězce: sourcing.	15
2.6	Příklad neúplné matice Vztahu: Úloha-Scénář (22).	17
3.1	Uzly a hrana instancí MBI objektů z prostředí Neo4j.	20
3.2	Grafická vizualizace modelu Labeled property graph [1].	21
3.3	Grafické prostředí databáze Neo4j.	22
3.4	Jednoduchý dotaz jazyka Cypher.	23
4.1	Staré schéma datového úložiště reprezentované dvěma entitami Uzel a Vztah [2].	26
5.1	Nové schéma datového úložiště reprezentované dvěma entitami Uzel a Vztah.	31
5.2	Příklad nového schématu datového úložiště.	33
5.3	Ukázka Uzlu METRIC a Vztahu TASK_METRIC.	34
5.4	Ukázka části modelu - hierarchická struktura Metriky.	36
5.5	Ukázka části modelu - okolí instance: „Počty spravovaných technických prostředků“.	36
6.1	Životní cyklus dotazu.	39
6.2	Metoda Specificky orientované dotazování.	41
6.3	Metoda Dotazování pomocí vazeb.	43
6.4	Průběh části Struktura dotazu.	44
6.5	Průběh části Specifikace dotazu.	45

7.1	Schématická část ontologie [3].	52
7.2	Datová část ontologie [3].	53
7.3	Příklad dotazu jazyka GQL [3].	54
7.4	Přeložený dotaz z obrázku 7.3 do jazyka SPARQL [3].	54
7.5	Příklad dotazu jazyka GQL s „kontextovým rámem“ [3].	55
8.1	Model nasazení aplikace pro dotazování.	58
8.2	Use-case model dotazovací aplikace.	59
8.3	Návrh koncepce dotazovacího nástroje (aplikace).	60
8.4	Schéma ke vzorovému dotazu číslo 4.	63
8.5	Výsledek dotazu číslo 4.	64

Úvod

Motivace

Na světě existuje mnoho standardů a metodik pro řízení podnikové informatiky. Jednou z nich, nabízených zejména pro české uživatele Katedrou informačních technologií na Vysoké škole ekonomické v Praze, je model, který se nazývá „Řízení podnikové informatiky“ (Management Byznys Informatiky, MBI) [4].

Pro práci s modelem byl vytvořen portál MBI obsahující řešení v rozvoji a řízení IT, respektive podnikové informatiky [5], do kterého mají všichni uživatelé bezplatný přístup. Hlavním smyslem jeho využívání je poskytnutí základních informací právě z oblasti podnikové informatiky primárně pro zaměstnance menších a středních firem, ale i pro ostatní. Získávání dat probíhá z hlediska informatiky probíhá získávání dat pomocí dotazování se nad několika objekty a jejich vazbami, které dohromady tvoří kompletní metamodel MBI.

Hlavním smyslem práce autora je zanalyzování potřeb uživatele pro dotazování se nad objekty MBI a z analýzy vyplývající návrh dotazovacího jazyka nad novým schématem datového úložiště. Tento jazyk by měl ve finální fázi poskytnout uživatelům alternativní způsob dotazování.

Práce volně navazuje na bakalářskou práci: „Vizualizace vztahů v informační bázi“ [2], se kterou se seznámíme v kapitole 1 a jejímž autorem je Bc. Vojtěch Stránský.

Struktura práce

Kompletní struktura práce je postavena na následujících kapitolách:

První kapitola (viz 1) popisuje MBI a bakalářskou práci: „Vizualizace vztahů v informační bázi“.

Ve druhé kapitole (viz 2) jsou provedeny analýzy potřeb uživatelů. Je zde rozebráno, proč a na co se chce uživatel ptát. Dále je v kapitole popsán současný stav portálu MBI a způsoby dotazování, které aktuálně nabízí. Rozebrány jsou i výhody a nevýhody portálu. Kapitola je zakončena souhrnem všech těchto analýz a nastíněním návrhu řešení, které z nich plyne.

Třetí kapitola (viz 3) se zaměřuje na cílové technologie: grafovou databázi Neo4j a dotazovací jazyk Cypher. Tyto technologie na první pohled možná nesouvisí s tématem, nicméně jsou pro čtenáře důležité k pochopení souvislostí.

Čtvrtá kapitola (viz 4) se zabývá analýzou současného schématu datového úložiště, které bylo již vytvořeno.

Pátá kapitola (viz 5) kompletně popisuje a prezentuje nové schéma datového úložiště. Tato kapitola přímo navazuje na kapitolu čtvrtou (viz 4), po jejím prostudování by čtenářům mělo být zcela jasné, jaké rozdíly obsahuje nové schéma oproti starému.

V šesté kapitole (viz 6) je popsán samotný návrh dotazovacího jazyka nad datovým úložištěm definovaným v kapitole 5. Návrh opět vychází z prováděných analýz.

Sedmá kapitola (viz 7) porovnává vzniklý dotazovací jazyk s jiným doménově specifickým jazykem, který je v krátkosti představen.

Poslední kapitola (viz 8) se zabývá implementační částí. Je v ní uvedeno, jakým způsobem je navržena implementace dotazovacího jazyka. Na samotném konci je provedena metoda „proof-of-koncept“.

Poslední část textu se věnuje vyhodnocení celé práce. Mimo jiné se zde objevuje diskuse o budoucnosti problematiky.

MBI a související práce

Pro zasvěcení do problematiky je nutné se seznámit s MBI a získat tak základní přehled o jeho fungování, významu a struktuře. Poslední část kapitoly se zabývá bakalářskou prací Bc. Vojtěcha Stránského, která nese název „Vizualizace vztahů v informační bázi MBI“ a souvisí s touto prací.

1.1 MBI - Základní informace

1.1.1 Definice MBI

„MBI (Management Byznys Informatiky) je portál obsahující zobecněná řešení v řízení provozu a rozvoje IT, resp. podnikové informatiky. Jeho smyslem je sdílet a využívat znalosti a doporučení vyplývající z praxe a z dalších zdrojů.“ [5]

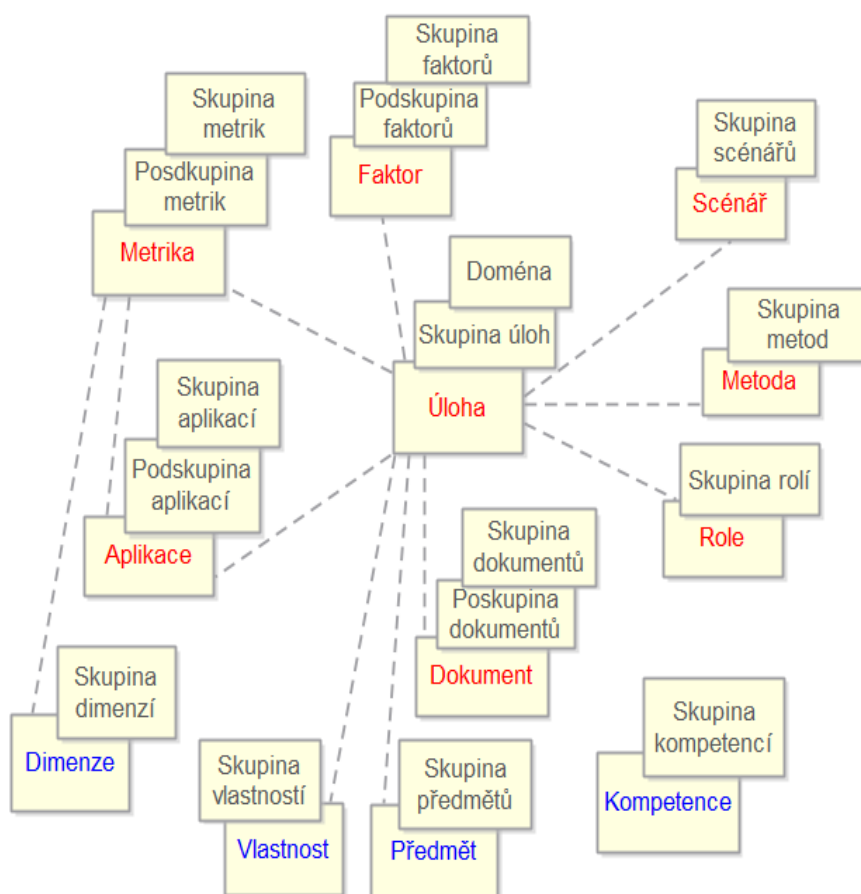
Následující podkapitoly čerpají informace ze zdrojů [4] a [5].

1.1.2 O portálu

Portál MBI byl vytvořen Katedrou informačních technologií na Vysoké škole ekonomické v Praze, za účelem poskytnout informace o podnikové informatice širší veřejnosti. V současné době má přístup do portálu, dostupný na webových stránkách ¹, každý uživatel. Vyžadována je pouze bezplatná registrace. Po ní je nám umožněn přístup do informační báze MBI.

Hlavním účelem portálu je nabídnout všem pracovníkům, kteří řídí informatiku v různých podnicích, flexibilní a konzistentní metodický rámec řízení podnikové informatiky a nejlepší praktiky z tohoto oboru - „best practice“. Podmínkou však je, že tyto firmy nevyužívají IT jako hlavní záměr podnikání. Pro ty je řízení informatiky postaveno na jiných principech.

¹<http://mbi.vse.cz/>



Obrázek 1.1: Model informační báze MBI.

Portál je stále ve vývoji. Nejen tým pracovníků MBI, ale také celá MBI komunita, se snaží o rozšiřování a vylepšování současného stavu. Doporučení, jak se samotným portálem pracovat, lze najít na odkazu ².

1.2 Struktura modelu - objekty MBI

Informační báze MBI je postavena na modelu dvanácti objektů, ze kterých uživatel získává data. Tyto objekty můžeme dále rozdělit na hlavní a podpůrné objekty. Každý z nich má navíc svou hierarchickou strukturu.

Všechny objekty i s jejich propojením můžeme vidět na obrázku 1.1. Podle vazeb pak uživatel dostává komplexní přehled nad tím, jak jsou mezi sebou objekty pospojovány. Uživatel by si měl také intuitivně odvodit význam těchto vazeb.

²<http://mbi.vse.cz/mbi/files/help.pptx>

Text k následujícím podsekcím byl čerpán ze zdroje [6].

1.2.1 Hierarchie objektů

Ještě před nedávnou dobou byla víceúrovňová hierarchie pouze u hlavních objektů, nyní se vyskytuje i u objektů podpůrných.

Každý objekt má dvouúrovňovou, nebo tříúrovňovou hierarchickou strukturu v závislosti na jeho typu. Nejnižší úroveň je dána právě tímto objektem a nese vždy jeho název. Některé objekty postrádají druhou úroveň, kterou označíme jako Podskupina. Nejvyšší úroveň pak nazveme Skupinou. Všechny vrstvy obsahují instance.

Případy názvů hierarchií (Skupin a případně i Podskupin) jsou uvedeny u každého objektu dále v textu, stejně jako konkrétní instance těchto úrovní.

1.2.2 Hlavní objekty

1.2.2.1 Úloha

Úloha funguje jako „centrální“ objekt, jelikož jsou na ni vázány všechny ostatní objekty (kromě Kompetence) a slouží tak jako hlavní přístupový bod do modelu. Úlohy poskytují uživateli informace a postupy řešení konkrétních problémů, na které se ptá.

Příklady instancí: Návrh IT služby, Řízení komunikace vedení IT s byznysem.

Skupina: Doména (Strategické řízení IT, ...); Podskupina: Skupina úloh (IT jako součást byznysu, ...).

1.2.2.2 Scénář

Scénář obsahuje informace o tom, jak se zachovat v konkrétních situacích nebo při řešení nějakého problému. Všechny tyto principy a postupy jsou ve Scénáři již definovány. Tento objekt lze chápat jako náhradní přístupový bod, a to hlavně kvůli své informační struktuře. Na Scénář se totiž váží doporučená řešení v rámci úloh MBI.

Příklady instancí: Je nutné vytvořit kvalitní katalog IT služeb, Současné IT aplikace a technologie neodpovídají potřebám podniku.

Skupina: Skupina scénářů/Skupina životních situací (Scénáře v řízení IT služeb, ...); Podskupina: Není.

1.2.2.3 Faktor

Faktor ukazuje na nějaký předmět z prostředí informačních technologií, na který se váže daná úloha.

Příklady instancí: Faktor velikosti podniku, Úroveň a formy sourcingu.

Skupina: Skupina faktorů (Byznys prostředí, ...); Podskupina: Podskupina faktorů (Velikost podniku, ...).

1.2.2.4 Role

Role určuje typy pracovních pozic z prostředí podniku, které mají vazby na úlohy. Členění role je postaveno na základě její kompetence – princip RACI (Responsible, Accountable, Consulted, Informed).

Příklady instancí: Vývojář softwaru, Tester.

Skupina: Skupina rolí (Vývojáři, ...); Podskupina: Není.

1.2.2.5 Dokument

Dokument poskytuje uživateli data, která jsou v různých formách a podobách, o vstupu nebo výstupu objektu Úloha.

Příklady instancí: Plán údržby, Projektový záměr.

Skupina: Skupina dokumentů (Dokumenty řízení IT služeb, ...); Podskupina: Podskupina dokumentů (Dokumenty plánování služeb, ...).

1.2.2.6 Metrika

Metrika nabízí data získaná měřením, která se využívají k řízení a správě podniku. Jsou k ní vázány další související metriky a analytické dimenze, které slouží k identifikaci a hodnocení sledovaných ukazatelů.

Příklady instancí: Objem spravovaných datovýchází v GB, Počet poskytovaných IT služeb.

Skupina: Skupina metrik (Metrika řízení IT zdrojů, ...); Podskupina: Podskupina metrik (Metriky datovýchází a jejich kvality, ...).

1.2.2.7 Aplikace

Aplikace navrhuje a popisuje všechny možné aplikace, které jsou vhodné pro podporu řízení IT.

Příklady instancí: Analýzy řešených IT projektů, Strategie sourcingu.

Skupina: Skupina aplikací (Analytické a plánovací aplikace pro podporu řízení IT, ...); Podskupina: Podskupina aplikací (Analytické a plánovací aplikace pro řízení rozvoje IT, ...).

1.2.2.8 Metoda

Metoda popisuje osvědčené a formalizované postupy a návody, jak správně realizovat cíle a postupy příslušné úlohy.

Příklady instancí: Daňové účetnictví, Metody řízení investic.

Skupina: Skupina metod (Metodiky a metody řízení podniku, ...); Podskupina: Není.

1.2.3 Podpůrné objekty

Podpůrné objekty nehrají v modelu MBI takovou roli. Uživateli většinou postačí data z hlavních objektů, od toho je použito v názvu slovo „podpůrné“.

1.2.3.1 Dimenze

Dimenze představuje analytická hlediska pro hodnocení a sledování jednotlivých ukazatelů.

Příklady instancí: Časová dimenze, IT projekty.

Skupina: Skupina dimenzí (Dimenze ekonomiky IT, ...); Podskupina: Není.

1.2.3.2 Vlastnost

Vlastnost reprezentuje vlastnost informatiky (informačního systému), kterou úloha ovlivňuje.

Příklady instancí: Bezpečnost, Výkonnost podnikové informatiky.

Skupina: Skupina vlastností (Vlastnosti podnikové informatiky a jejich komponent, ...); Podskupina: Není.

1.2.3.3 Předmět

Předmět představuje hlavní předměty řízení, které jednotlivé úlohy ovlivňují. V portálu jsou instance objektu Předmět zastupovány názvem Objekt řízení.

Příklady instancí: Organizace ICT útvaru, Podnikové procesy.

Skupina: Skupina předmětů/Skupina objektů řízení (Předměty řízení podnikové informatiky, ...); Podskupina: Není.

1.2.3.4 Kompetence

Představují přehled standardních kompetencí pracovníků ve vztahu k podniku. Stávají se součástí objektu Role.

Příklady instancí: Návrh a vývoj aplikací, Schopnost komunikace.

Skupina: Skupina kompetencí (Technické kompetence, ...); Podskupina: Není.

1.3 Vizualizace vztahů v informační bázi MBI

V této podkapitole jsou používány informace ze zdroje [2].

1.3.1 Souhrn práce

„Vizualizace vztahů v informační bázi MBI“ [2] je název bakalářské práce Bc. Vojtěcha Stránského, která vznikla v roce 2014. Jejím hlavním cílem bylo vytvořit softwarový systém, který umožní transformovat data z MBI a vytvořit

novou grafovou databázi Neo4j³. Pro procházení databáze vytvořil autor práce pár předpřipravených dotazů v jazyce Cypher.

Transformace dat byla provedena z XML dokumentu, který dodává strana MBI. Tento dokument obsahuje data o objektech a jejich vztazích. Ze zmiňovaného XML souboru a s pomocí programu v jazyce Java, který autor sám vytvořil, vznikají CSV soubory, ze kterých následně probíhá import dat z modelu MBI do databáze Neo4j. Při tvorbě práce bylo definováno staré schéma datového úložiště.

O grafových databázích, Neo4j a jazyku Cypher dále pojednává kapitola 3. O tom, jak bylo definováno výše zmiňované schéma, se dozvíte v kapitole 4.

1.3.2 Přínos práce

Pro MBI měla práce pana Stránského poměrně velký přínos. Grafová databáze Neo4j nabízí uživatelům zcela odlišný způsob zobrazování výsledků dotazů. Dalším přínosem, který je jen nastíněn, je dotazování se pomocí připravených dotazů v jazyce Cypher.

Pro moji práci měla „Vizualizace vztahů v informační bázi“ důležitý přínos, jelikož bylo definováno schéma datového úložiště, ze kterého vychází nové schéma, popsané v kapitole 5. S pomocí transportu dat z XML dokumentu a následného importu CSV souborů do databáze Neo4j bylo úložiště naplněno daty. Z dalšího hlediska byla nastíněna možnost vytvořit nový dotazovací jazyk pro MBI objekty, čili námět pro tuto bakalářskou práci.

³<http://neo4j.com/>

Analýza uživatelských potřeb

Před návrhem samotného schématu a jazyka je důležité provést analýzu, ze které budeme vycházet. Nejprve se zaměříme na vymezení nejčastějších potřeb uživatelů portálu, tj. na to jakým způsobem a na co se dotazují. Poté se podíváme na současný stav portálu a na způsoby dotazování, které jsou uživatelům nabízeny. Rozebrány budou také výhody a nevýhody portálu. Ve finální fázi zohledníme probrané analýzy a dospějeme k závěrečnému rozhodnutí o dalším postupu.

2.1 Uživatelé a jejich dotazy

Jako první krok je důležité ujasnit si, kteří uživatelé používají bázi MBI pro získávání dat a jaké typy dotazů nejčastěji kladou.

2.1.1 Analýza uživatelů

Řízení podnikové informatiky může být postaveno na různých principech. Informační báze MBI je zaměřena tak, aby její uživatelé nevyužívali IT ve svých firmách jako hlavní záměr podnikání [4]. Lidé z těchto firem, vyjma manažerů v oboru podnikové informatiky, by měli být nejčastějšími uživateli portálu MBI, jelikož pro ně je primárně určen. Nutno podotknout, že mezi těmito firmami se vyskytují převážně ty, které jsou malého či středního vzrůstu, myšleno na počet zaměstnanců. Větší firmy využívají jinak zaměřené metodiky (ITIL atd.).

Druhou sortou lidí, kterou by měl portál přilákat, je právě skupina potenciálních manažerů v oboru podnikové informatiky [4]. Těm MBI nabízí dobrý zdroj informací k jejich profesnímu vývoji.

Do poslední skupiny můžeme zařadit všechny ostatní uživatele. Ať už se jedná o členy týmu MBI, studenty, či jen zájemce o tento obor, všichni mají do báze volný přístup. Hlavní pozornost by měla být zaměřena na výše zmiňované skupiny.

2.1.2 Uživatelské dotazy

Po přečtení předchozí analýzy dospějeme k závěru, že nejvíce dotazů by mělo směřovat od firemních zaměstnanců (bez manažerů), kteří se chtějí dozvědět nějaké informace z oblasti podnikové informatiky. O jaké informace by se mělo jednat, záleží na problému nebo úkolu, který právě řeší nebo mají vyřešit.

Obecně, pokud dostaneme zadán úkol, snažíme se najít co nejrychlejší a nejefektivnější řešení. Informace, které můžeme získat při jeho hledání, avšak nesouvisející s úkolem samotným, pro nás nemusí být zbytečné, ale vzhledem k danému problému jsou irelevantní. Dotazy uživatelů ze zmiňovaných firem tedy v převážné míře povedou striktně k hledání konkrétních odpovědí. Takový uživatel se totiž chce ptát na věci, které mu umožní najít rychlé řešení.

Informační báze MBI obsahuje velké množství dat, rozdělených do objektů MBI, popsanych v podkapitole 1.2. Není tak lehké určit, jaké objekty uživatel při dotazování využije. Jediným spolehlivým faktorem, kterého se můžeme držet, je struktura modelu.

Za nejčastější dotazy můžeme brát ty, které se týkají objektů Úloha a Scénář. Proč tomu tak je, se dozvíme dále v kapitole v části 2.2.1, ale již při pohledu na obrázek modelu 1.1 je zřejmé, že je to hlavně u Úlohy kvůli jejím vazbám. Naopak mezi méně používané dotazy patří ty, které jsou orientovány na podpůrné objekty, které byly rozebrány v části 1.2.3. To je zapříčiněno jejich doplňujícím významem pro model.

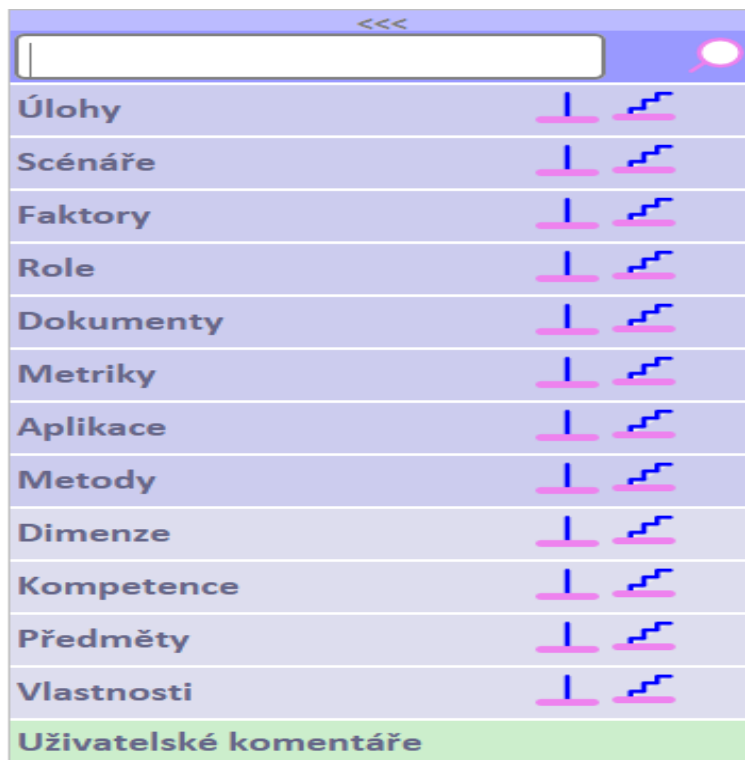
Na konci této podkapitoly můžeme dospět k závěru, že typy dotazů by neměly být moc složité. Mělo by se jednat převážně o dotazy jednoduchého druhu, které jsou jasně specifikovány a uživateli by měly poskytnout postačující a rychlou odpověď. Na druhou stranu, portál MBI je přístupný všem a určitě se najde velké množství takových, kteří mají zájem o „brouzdání“ modelem a zjišťování celkového přehledu o vztazích mezi objekty (instancemi objektů). Proto je nutné v budoucím návrhu zohlednit i tuto variantu, která by mohla být velice zajímavá.

2.2 Analýza současného stavu portálu

2.2.1 Přístup k objektům

Aktuálně je na portálu MBI doporučované přistupovat k funkcím a datům skrze objekty třemi možnými variantami [6]:

- Prvním způsobem je přístup skrze objekt Úloha, který má vazby téměř na všechny ostatní objekty z modelu a poskytuje tak vhodný výchozí bod. Tento postup je zřejmě nejpoužívanější na celém portálu, k čemuž přispívá i významová stránka Úlohy.
- Druhou doporučovanou možností je prostřednictvím Scénáře. Uživatel by měl mít dobře specifikovaný svůj problém, na který se chce dotázat,



Obrázek 2.1: Postranní menu po přihlášení do portálu.

a podobný pak najít v tomto objektu. S pomocí ostatních vazeb, hlavně na Úlohu, pak není složité získat požadované odpovědi.

- Posledním způsobem je přístup přes konkrétní objekt z modelu. Uživatel si může vystačit s daty ze samotného objektu, nebo znovu za pomoci vazeb k Úloze čerpat další data. V případě některých objektů může být využito i jiných vazeb, viz obrázek 1.1.

Všechny zmiňované varianty by měly být známé všem uživatelům, aby dle svého uvážení mohli zvolit nejlepší postup, jak k datům přistoupit (jak se dotazovat).

2.2.2 Rozbor dotazovacích způsobů

V následujících částech textu se dozvíme něco o způsobech dotazování, které jsou zpřístupněny návštěvníkům portálu.

Na obrázku 2.1 je vidět postranní menu, které se zobrazí na úvodní stránce po přihlášení do portálu. Uživatelům slouží jako nástroj pro dotazování se pomocí jednoduchého klikání na grafické prvky. Ve výčtu objektů můžeme vidět po pravé straně dva malé obrázky. První z nich (více nalevo) připomíná

Úlohy			
Úloha			
	0.7		U001A : Podpora IT cílům byznysu vlastního podniku
	0.6		U002A : Zajištění znalostí IT o podnikové ekonomice
	0.5		U003A : Řízení kooperace IT s byznysem na strategii byznysu
	0.8		U004A : Revize IT strategie dle požadavků byznysu
	0.8		U005A : Řízení komunikace vedení IT s byznysem
	0.7		U005B : Spolupráce manažerů byznysu a IT - CIO a CMO
	0.6		U006A : Spolupráce IT na tvorbě byznys modelu
	0.6		U007A : Provozní model byznysu

Obrázek 2.2: Výpis instancí objektu Úloha.

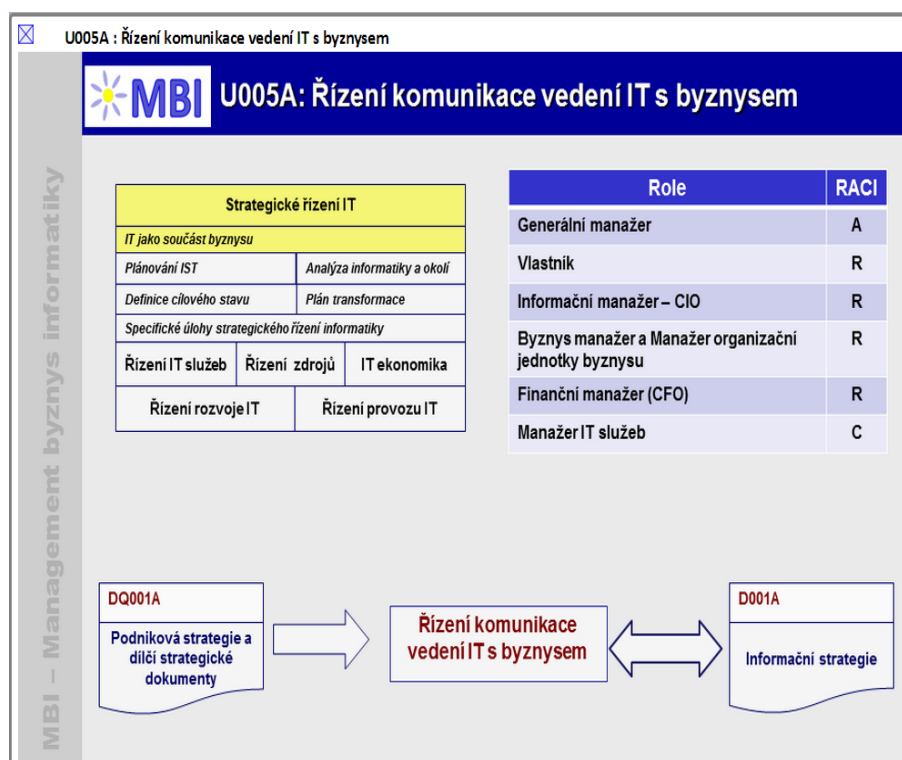
svislou modrou čáru a po jeho „rozkliknutí“ přistoupíme přímo na objekt. Druhý obrázek, mající schodovitý tvar, umožňuje přístup k objektu podle hierarchických úrovní. Nyní si popíšeme následující způsoby podrobněji.

2.2.2.1 Dotazování přes objekt

Jak již bylo zmíněno, uživatel může jako jednu možnost k přístupu do modelu zvolit výběr konkrétního objektu. Pro přímý přístup k jeho instancím nám slouží jeden ze dvou malých obrázků vyskytující se ve výčtu objektů více nalevo.

Po zvolení této metody se nám zobrazí seznam všech instancí vybraného objektu evidovaných v databázi. Jak můžeme vidět na obrázku 2.2, jedná se v tomto případě o objekt Úloha. Pro další postup můžeme znovu vybírat z více variant.

- Prvním způsobem je kliknutí na ikonku s třemi obdélníky, která uživateli zobrazí základní data o objektu (tzv. souhrnný slide). Každý slide se liší v závislosti na vybraném objektu. Tento způsob je dobrou možností pro uživatele, jak se lépe zorientovat v hierarchii či struktuře



Obrázek 2.3: Souhrnný slide k instanci: „Řízení komunikace vedení s IT byznysem“.

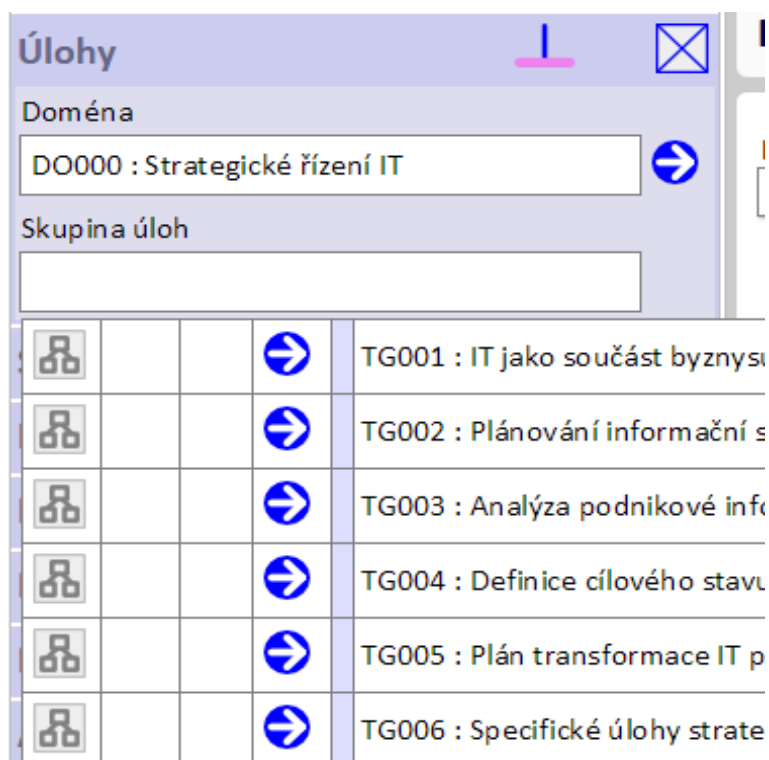
konkrétního objektu a pomáhá mu k tomu, aby si ho lépe představil v celkovém modelu s jeho obecnými vlastnostmi.

Na našem příkladě na snímku 2.3 jsme demonstrovali souhrnný slide k instanci objektu Úloha, konkrétně instanci s názvem: „Řízení komunikace vedení s IT byznysem“. Všimněme si, že na obrázku jsou zobrazena data o rolích, i se vztahem RACI, uvedeném v části 1.2.2.4, které se k této instanci vážou.

- Jako druhou možnost můžeme vybrat zobrazení dat instance objektu. Zobrazení se uskuteční přes bílou šipku v modrém kruhu na obrázku 2.2. Na portálu se poté objeví přehled vazeb objektu, pomocí kterých se dá dále pokračovat na konkrétní vazby, které jsou v současnosti zobrazovány maticemi. Pod těmito vazbami jsou data o samotné instanci (cíl, popis, obsahová vymezení a další).

2.2.2.2 Dotazování přes hierarchie

Dotazování přes hierarchickou strukturu je druhou variantou, jak se pohybovat v modelu a dostat se tak k datům. K tomuto postupu se dostaneme přes ikonu



Obrázek 2.4: Ukázka dotazování přes hierarchickou strukturu objektu Úloha.

schodovitého tvaru, v základním (postranním) menu napravo.

Metoda tohoto dotazování je podobná metodě popsané v části textu 2.2.2.1, nicméně je rozsáhlejší. V prvním kroku je totiž nutné vyplnit Skupinu daného objektu, na našem obrázku 2.4 se znovu jedná o Úlohu. Vidíme, že v prvním kroku byla vybrána Doména: „DO000: Strategické řízení IT“. Doména byla zvolena podobným způsobem, jakým můžeme vybírat ze Skupiny úloh ve výčtu.

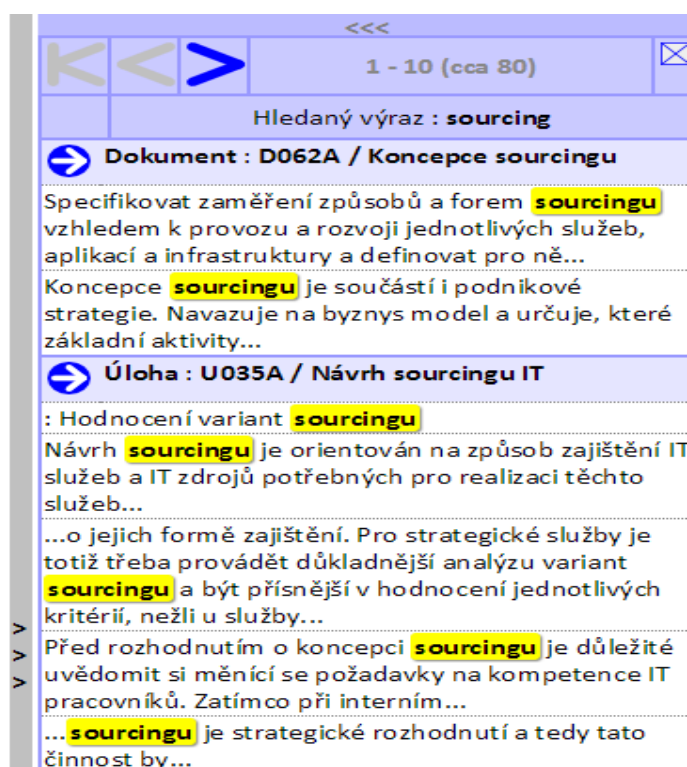
Kdybychom postupovali dále podle obrázku 2.4, vybrali bychom nějakou konkrétní instanci ze Skupiny úloh a dále bychom vybírali již instanci Úlohy, která spadá pod hierarchickou strukturu, kterou jsme tímto procházením zvolili. V mezikrocích nemusíme rovnou jednat podle uváděného postupu, ale můžeme například zobrazovat data vybraných instancí určité hierarchické úrovně dané instance a podobně.

2.2.2.3 Dotazování přes vyhledávání

V postranním menu z obrázku 2.1 se vyskytují dvě části, které v tomto rozboru ještě nebyly popsány a nemusely by být uživateli zcela jasné.

První z nich je text Uživatelské komentáře, který se nachází ve spodní

2.3. Co je dobré a co by šlo vylepšit?



Obrázek 2.5: Výpis výsledků vyhledávání řetězce: sourcing.

části. Možnost přidávat komentáře, jakožto přihlášený uživatel, patří mezi jednu z nových funkcí portálu a v tomto rozboru se jí nebudeme dále věnovat.

Druhou částí, pro nás více zajímavou, je část zcela nahoře. Můžeme zde vidět prázdné okno, které slouží pro fulltextové vyhledávání na portálu. Do okna může návštěvník psát libovolné řetězce. Pro vyhledání výsledku je nutné použít malou lupu na pravé straně. Po jejím stisknutí se zobrazí výpis výsledků, který můžeme vidět na obrázku 2.5.

Pro názorný příklad byl zvolen řetězec sourcing. Ve výpisu je umožněno posouvání v závislosti na počtu výsledků. Další možností je přistupování k instancím, se kterými je řetězec spojen.

2.3 Co je dobré a co by šlo vylepšit?

Před shrnutím celé analýzy a vysvětlením toho, co z ní vyplývá, se zaměříme na téma: „Co je dobré a co by šlo vylepšit?“ a popíšeme tedy výhody a nevýhody portálu. V analýze budou uváděny výhody a nedostatky na základě názoru autora.

2.3.1 Výhody portálu

Portál MBI, jak již bylo připomínáno, je stále ve vývoji a snaží se přidávat nová vylepšení, která zlepší jeho použitelnost i spokojenost uživatelů. Mezi jeho největší výhody patří postranní menu ze snímku 2.1. Menu reprezentuje hlavní výchozí bod pro dotazování se. Uživatel si může na začátku vybrat, jakým způsobem se při dotazování vydá, viz sekce 2.2.2.

Další výhodou je možnost práce s nalezenými daty. Po tom, co dokončíme dotazování, dostaneme určité množství dat. Portál v současnosti umožňuje stažení těchto dat do souboru typu PDF, vytvoření reportu v textovém procesoru Microsoft Word, nově také přidávat uživatelské komentáře k výsledným instancím a další užitečné doplňky.

Při tvorbě portálu a analýzy toho, jak by měl vypadat a nejefektivněji fungovat, je obtížné najít řešení, které bude vyhovovat všem jeho uživatelům. Je proto nutné počítat s odezvou uživatelů a na jejím základě portál upravovat.

2.3.2 Nedostatky portálu

Nedostatky portálu byly rozděleny do dvou částí, dvou hlavních nedostatků. Po jejich přečtení se dozvíme, že spolu navzájem souvisí, nicméně jsou uvedeny odděleně.

2.3.2.1 Problém nepřehlednosti

Za první nevýhodu můžeme vybrat částečnou nepřehlednost v některých krocích při dotazování se nad objekty. Jedná se převážně o finální stav, kdy má uživatel zobrazenou konkrétní instanci a je složité se zorientovat ve velkém množství dat (u „velkých instancí“). S problémem souvisí i tabulka s vazbami k ostatním objektům. Po kliknutí na konkrétní typ vazby, například Vztah: Úloha-Scénář (22), se objeví nová stránka s maticí, která je hlavně u velkých matic nepřehledná. O problému matic se dozvíme dále v části 2.3.2.2.

Pro některé uživatele by mohlo být matoucí dotazování přes hierarchie, popsáno v textu 2.2.2.2. Uživatel by si měl po čase na tuto variantu zvyknout a zorientovat se v ní. I přes tuto skutečnost je důležité zvážit možné nebezpečí. Vzhledem ke způsobu, jakým jsou hierarchie objektů navrženy, není s největší pravděpodobností možné docílit efektivnějšího řešení.

2.3.2.2 Problém matic

Jako názorná ukázka problému matic byla vybrána matice popisující vztah instance objektu Scénář s názvem: „IT funguje rutinně, nepřispívá ke zvyšování výkonnosti podniku“.

Pro zobrazení matice vazeb, kterou si můžeme prohlédnout na snímku 2.6, bylo potřeba zvolit Vztah: Úloha-Scénář (22) z tabulky vztahů této instance. Číslo 22 uváděné ve Vztahu udává počet vazeb instance Scénáře k instancím

Vztah: Úloha-Scénář (22)											
Jen existující vazby? <input checked="" type="checkbox"/>											
Typ: Význam úlohy		S001	S002	S003	S004	S005	S006	S031	S032	S033	S034
Úloha:											
Podpora IT cílům byznysu vlastního podniku	U001A	3	3	2	3	1		2	3		
Řízení kooperace IT s byznysem na strategii byznysu	U003A	1	2	2				2	2	3	
Revize IT strategie dle požadavků byznysu	U004A	2		3				3	3	3	
Spolupráce IT na tvorbě byznys modelu	U006A	3	2	1				1	2	3	

Obrázek 2.6: Příklad neúplné matice Vztahu: Úloha-Scénář (22).

Úlohy vypsány v řádcích. Na první pohled poznáme, že matice není na snímku vyobrazena celá. Vazby Scénáře: „IT funguje rutinně, nepřispívá ke zvyšování výkonnosti podniku“ jsou ve sloupci, který je zvýrazněný žlutou barvou. Okolní sloupce reprezentují instance objektu Scénář, které spadají do stejné Skupiny.

Zobrazená matice působí na uživatele nepřehledným dojmem, který vytváří její struktura a vizualizace na stránce. Tento dojem se zvyšuje v závislosti na velikosti počtu vztahů.

2.4 Výsledek analýzy

V této kapitole jsme zanalyzovali věci, které se týkají uživatelských potřeb. Rozdělili jsme uživatele do třech hlavních skupin. Dále jsme popsali aktuální stav portálu MBI - doporučení, jak by měli uživatelé přistupovat k objektům a tři dotazovací způsoby, podle kterých se uživatelé ptají. Před touto částí proběhla krátká analýza na téma: „Co je dobré a co by šlo vylepšit?“.

Z proběhlé analýzy uživatelských potřeb vyvodíme některé závěry:

- Není potřeba upravovat metody dotazování na portálu MBI. Jejich ře-

šení obsahuje několik nedostatků, nicméně bude vhodnější vytvořit nové schéma datového úložiště, dále v kapitole 5, a na něm postavený nový dotazovací jazyk, rozebraný v kapitole 6.

- Na základě předchozího bodu vyplyne hlavní myšlenka práce, poskytnout uživatelům alternativní způsob dotazování, který bude navíc využívat nových technologií, popsaných v kapitole 3, mimo jiné s velice přívětivou vizuální stránkou. Tyto technologie by měly vyřešit problématic, popsaný v části textu 2.3.2.2.
- Nový dotazovací jazyk bude vycházet z analýzy uživatelů a jejich dotazů, viz sekce 2.1. Při jeho tvorbě by měla být v první řadě zohledněna možnost, kdy se uživatel chce dotazovat pouze na samotný objekt. Dále by měla být zvažena varianta, kdy se uživatel zajímá o širší kontext.

Cílová technologie

V první kapitole 1 jsme se seznámili s pojmem MBI, ze kterého budeme vycházet. Pro komplexní zasvěcení do problematiky je nutné získat alespoň základní informace o cílové technologii. K tomu by nám měla postačit právě tato kapitola, která se úvodem zaměří na grafové databáze. Mezi tyto databáze patří i Neo4j, které bude věnována velká část. Nakonec si představíme dotazovací jazyk Cypher, související s těmito technologiemi.

3.1 Grafové databáze a Neo4j

3.1.1 Grafové databáze

Kromě relačních databází existují i jiné databáze, které je dobré znát a je s nimi třeba v budoucnu počítat. Mezi tyto databáze můžeme zařadit právě i ty grafové.

3.1.1.1 Definice grafové databáze

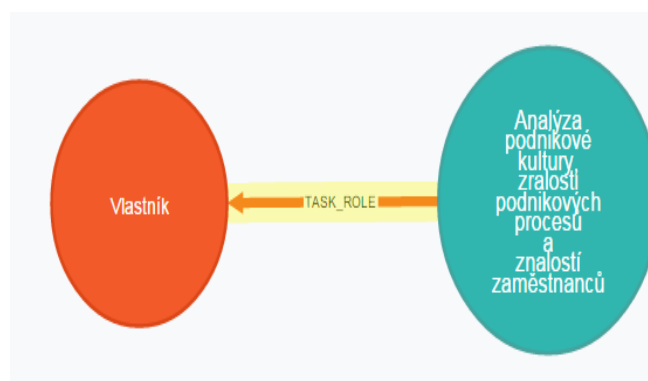
Grafová databáze, jejímž datovým modelem je graf, přesněji tzv. „Property graph“⁴, je typ NoSQL databáze⁵ [7].

Grafové databáze tedy využívají dvou základních druhů entit, uzlů a hran. Obě tyto entity mají své speciální vlastnosti, které nebudeme dále rozebírat. To nejdůležitější, co bychom si měli pamatovat, je přítomnost uzlů, které uchovávají data nějaké entity. Hrany pak spojují vždy dva uzly pomocí vztahu. Dotazování se nad takovými daty poté využívá algoritmy známé z teorie grafů [8].

Pro představu definice a ukázkou spojitosti grafových databází s MBI modelem si můžeme prohlédnout uzly a hranu na obrázku 3.1. Obrázek byl nasnímán z prostředí Neo4j, viz text 3.1.3. Na snímku můžeme vidět instanci

⁴Property graph se skládá z uzlů a hran. Každý uzel a hrana mohou mít libovolné množství atributů (uspořádaných dvojic <klíč,hodnota>). Hrany mohou být orientované nebo neorientované.

⁵<http://nosql-database.org/>



Obrázek 3.1: Uzly a hrana instancí MBI objektů z prostředí Neo4j.

objektu Úloha: „Analýza podnikové kultury, znalosti podnikových procesů a znalostí zaměstnanců“; instanci objektu Role: „Vlastník“. Tyto dvě instance reprezentují uzly grafové databáze. Hrana je pak popsána vztahem: „TASK_ROLE“. O těchto vztazích se dozvíme více v kapitolách 4 a 5.

3.1.1.2 Výhody a nevýhody grafových databází

Mezi hlavní výhody grafových databází patří lehčí vyjádření mnohých dat, které by bylo složité dostat například do známých relačních databází (MySQL atd.). Tyto databáze jsou také vhodné pro uchovávání nepravidelných a komplexních dat [9].

Velkou nevýhodou je cena za vztah. Při zápisu dat do grafové databáze musíme čekat více času, než při zápisu dat do relačních a jiných databází, čili zápis velkých dat může být pomalý [8]. Relační databáze oproti grafovým databázím neumožňují efektivní uložení a dotazování grafových dat [10].

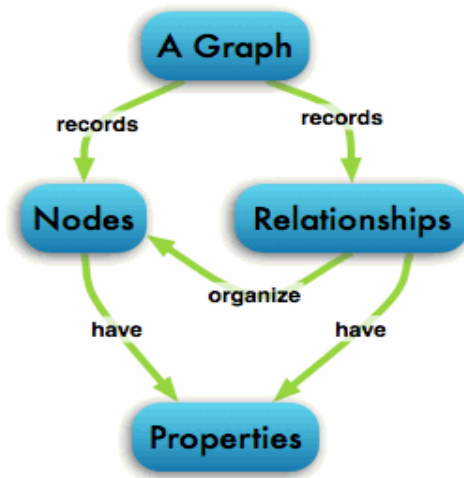
3.1.1.3 Zástupci

Grafové databáze mají několik hlavních zástupců. V současné době patří tyto databáze mezi jednu z technologií, na kterou je orientována velká pozornost. Každá grafová databáze se ve většině případů liší svou strukturou, nicméně všechny vycházejí z teorie grafů [11].

Jednou z nejpopulárnějších databází ve své kategorii je Neo4j, které se budeme věnovat dále v textu. Dalšími zástupci grafových databází jsou například: SonesGraphDB, AllegroGraph, FlockDB, InfiniteGraph, OrientDB a další [12].

3.1.2 Neo4j

Pro ukládání dat z MBI byla vybrána grafová databáze Neo4j. Bude dobré dozvědět se o ní bližší informace.



Obrázek 3.2: Grafická vizualizace modelu Labeled property graph [1].

3.1.2.1 Úvod

Jak již bylo uvedeno, Neo4j patří mezi grafové databáze a spadá do NoSQL databází (Not Only SQL). Jedná se o open source řešení, které je implementováno v programovacím jazyce Java. První verze Neo4j byla vydána v roce 2007 a jejím autorem je společnost Neo Technology [11].

3.1.2.2 Labeled property graph

Neo4j používá datový model, kterému se říká Labeled property graph. Na rozdíl od modelu Property graph, popsáném v textu 3.1.1.1, má Labeled property graph speciální vlastnost, která umožňuje přiřadit uzlu libovolný počet tzv. Labels. Label, neboli nálepka, přiřazuje uzel do určité množiny. Každý vztah má nějaký typ, který dává vztahům mezi dvěma uzly význam. Všechny tyto vztahy jsou navíc směřované, čili musí mít počáteční a koncový uzel [13].

Pro lepší představu je tento model vizualizován na obrázku 3.2.

3.1.2.3 Proč Neo4j?

Neo4j byla vybrána jako nové úložiště dat MBI pro práci popsanou v podkapitole 1.3. V této práci nebudeme cílovou technologií měnit a zůstaneme u Neo4j.

Jednou z hlavních výhod Neo4j a obecně grafových databází je ve vizualizační síle a velké líbivosti pro uživatele. Tyto výhody by měly odstranit některé nedostatky na portálu MBI, uvedené v textu 2.3.2. MBI model také dobře zapadá do koncepce grafů, kde instance objektů jsou reprezentovány uzly a vztahy mezi objekty pomocí hran.



Obrázek 3.3: Grafické prostředí databáze Neo4j.

Všechny tyto prvky mohou být důležitým základem k tomu, aby spolu s navrhovaným jazykem utvořily více než jen alternativní způsob pro dotazování se na MBI objekty.

3.1.3 Prostředí Neo4j

Do prostředí Neo4j se přistupuje přes webový prohlížeč. Snímek, který toto prostředí stručně popisuje, je pořízen z aktuálně nejnovější verze Neo4j: 2.1.7. Můžeme ho vidět na obrázku 3.3.

Na levé straně se nachází postranní menu, které obsahuje informace o aktuální databázi Neo4j a pomocné informace. Pomocí menu je nám umožněno ukládat předpřipravené Cypher dotazy. V horní části se nachází editor pro psaní Cypher dotazů, které můžeme například vykonávat a ukládat pomocí tlačítek na pravé části v místě editoru. Poslední popsanou částí na snímku je stream pro zobrazování výsledků dotazů, které provádíme.

3.1.3.1 Práce s Neo4j

Poslední část textu o Neo4j, která získává informace ze zdroje [13], se zaměří na práci se samotnou databází a její procházení.

Procházení databáze probíhá následujícím způsobem: „spustíme se“ na vybranou část grafu. Toho můžeme docílit například vyhledáním konkrétního indexu jednoho z uzlů. Po jeho nalezení můžeme prozkoumávat okolí do volitelných směrů.

```
MATCH (a)-[:`TASK_ROLE`]->(b) RETURN a,b LIMIT 25
```

Obrázek 3.4: Jednoduchý dotaz jazyka Cypher.

Výhoda tohoto způsobu je, že i když se v databázi vyskytuje velké množství dat (až v řádech milionů), při prohledávání okolí pracujeme pouze s daty (uzly a hrany), která souvisejí s tímto uzlem. Uživatelé tak nejsou zbytečně zpomalováni.

Tato vlastnost grafové databáze Neo4j bude využita při návrhu dotazovacího jazyka.

Vyhledání uzlů a hran může být provedeno pomocí dotazů jazyka Cypher, který si popíšeme dále v textu.

3.2 Jazyk Cypher

Do cílové technologie můžeme zařadit jazyk Cypher, jelikož ho využívá grafová databáze Neo4j pro dotazování se. Druhým důvodem, proč si tento jazyk popíšeme, je využití Cypheru v návrhu vlastního dotazovacího jazyka, definovaného v kapitole 6.

3.2.1 O Cypheru

Cypher je deklarativní dotazovací jazyk, který je inspirovaný jazykem SQL a v menší míře se mu podobá. Umožňuje nám popisovat to, co chceme vybrat, vložit, upravit nebo vymazat z grafové databáze bez toho, aniž by po nás vyžadoval popis, jak toho docílit [14].

Jazyk Cypher byl vytvořen z důvodu zjednodušení dosavadních mechanismů dotazování v databázi Neo4j. Konkrétně se jednalo o Java API, které bylo moc „upovídáné“. V druhé řadě pak o jazyk Gremlin, který byl příliš „nařizující“ [15].

Kromě jazyka SQL je Cypher inspirován i jinými jazyky. Za zmínku stojí jazyk SPARQL, ze kterého si vypůjčuje tzv. „pattern matching“ [16].

3.2.2 Struktura dotazu

Dotazy jsou v jazyce Cypher postaveny na několika základních klauzulích. Klauzule jsou v dotazu mezi sebou řetězeny a předávají si mezi sebou okamžité výsledky. Pro příklad si uvedeme klauzuli MATCH, jejíž odpovídající identifikátory (výsledky) budou výchozím kontextem pro další klauzuli [16].

3. CÍLOVÁ TECHNOLOGIE

Další částí dotazu jsou vzory (patterns), podle kterých jsou vyhledávány uzly a hrany. Tyto vzory mohou mít různé tvary, například cyklické [17].

Klauzule a vzory společně vytvoří základní kostru dotazu. Pro jeho kompletní specifikaci se používají různá klíčová slova. Příklad jednoduchého Cypher dotazu můžeme vidět na obrázku 3.4. Tento dotaz vrací dvojice uzlů (symbolicky označených „a“ a „b“), které jsou propojené hranou typu TASK_ROLE. Dotaz vrací prvních pětadvacet takových dvojic.

Analýza starého schématu datového úložiště

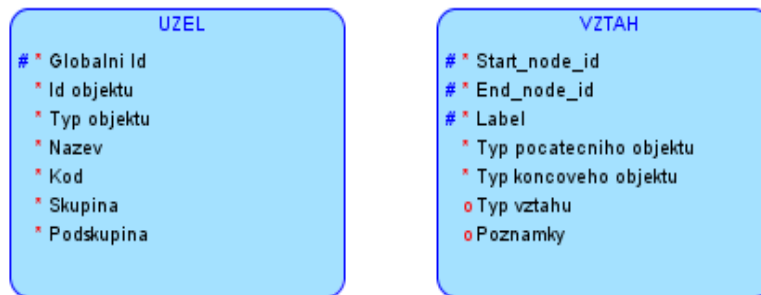
Tato krátká kapitola popisuje schéma datového úložiště, které bylo vytvořeno v bakalářské práci, která je stručně rozebrána v podkapitole 1.3. Navazující kapitola 5 pak bude ze schématu vycházet, proto bude vhodné se s ním seznámit.

Schématem datového úložiště je graf. Uzly a hrany mají své vlastní atributy, které budou v analýze popsány. V první části kapitoly se budeme věnovat atributům uzlu, v další části atributům hrany. V závěrečné části se dozvíme informace o tom, jak probíhal import dat do úložiště.

4.1 Struktura uzlů

Uzly ve schématu reprezentují objekty modelu MBI a jsou tvořeny následujícími atributy:

- **Globální Id** - určuje interní (uměle generovaný) identifikátor uzlu.
- **Id objektu** – identifikátor (číslo) instance jednoho z objektů MBI.
- **Typ objektu** – krátký string (řetězec znaků) popisující slovně objekt (TASK, METRIC a další).
- **Název** – slovní popis názvu instance.
- **Kód** – zkratka instance používaná na portálu. Začíná velkým písmenem/písmeny podle typu objektu, následuje několik číslic a může končit znovu velkým písmenem (U001A, UQ002A, R301 a další).
- **Skupina** – slovní popis názvu Skupiny.
- **Podskupina** – slovní popis názvu Podskupiny. Jestliže není vyplněna, obsahuje prázdný string.



Obrázek 4.1: Staré schéma datového úložiště reprezentované dvěma entitami Uzel a Vztah [2].

4.2 Struktura hran

Hrany ve schématu reprezentují vztahy mezi objekty MBI a jsou popsány atributy v tomto pořadí:

- **start_node_id** – identifikátor výchozího uzlu (Globální Id – uměle generovaný – interní).
- **end_node_id** – identifikátor koncového uzlu (Globální Id – uměle generovaný – interní).
- **label** – popis hrany ve formátu výchozí_koncový uzel, například APPLICATION_METRIC, někdy označován jako typ vazebního objektu.
- **Typ počátečního objektu** – typ objektu výchozího uzlu.
- **Typ koncového objektu** – typ objektu koncového uzlu.
- **Typ vztahu** - někdy označován jako Label. Udává název daného vztahu mezi dvěma uzly - počátečním a koncovým.
- **Poznámky** – textové poznámky o vztahu.

V analýze jsme si popsali dvě základní entity schématu (grafu): uzel a hranu. Těmto entitám byl přiřazen název Uzel a Vztah. Pro grafickou představu si je můžeme prohlédnout na snímku 4.1.

4.3 Import dat do úložiště

Jakožto datové úložiště byla vybrána grafová databáze Neo4j. Pro její vytvoření použil autor tohoto schématu (pan Stránský) program Batch importer ⁶.

⁶Program je dostupný na webových stránkách: <https://github.com/jexp/batch-import>

Tento program byl vytvořen pro importování dat pomocí CSV souborů.

Batch importer pracuje s dvěma CSV soubory. První soubor obsahuje uzly, druhý soubor hrany. Aby byl import dat do Neo4j úspěšný, bylo nutné dodržet danou strukturu těchto souborů. První řádka CSV souboru popisuje atributy svými názvy v sloupcích. Atributy jsou odděleny pomocí tabulátoru. Další řádky obsahují samotná data uzlů a hran, v tomto případě data entit Uzel a Vztah. Před samotným importem došlo ještě k úpravám CSV souborů pomocí programu v jazyce Java, které měly provést například přeindexování uzlů (přiřadit jim unikátní identifikátory). Úpravy bylo nutné provést i po importu dat do Neo4j, jelikož program Batch importer neumožňuje přiřadit uzlu jeho typ. Pro tyto úpravy pomohl autorovi jazyk Cypher, který Neo4j používá [2].

Princip importu dat a definování schématu datového úložiště využijeme i v této práci při tvorbě nového návrhu.

Návrh nového schématu datového úložiště

Před návrhem dotazovacího jazyka je potřeba definovat nové schéma datového úložiště, ze kterého budeme vycházet. V samotném úvodu vysvětlíme, proč zavádíme nové schéma. V další části schéma popíšeme a definujeme. Poslední část textu bude zaměřena na prezentaci tohoto schématu v praxi, čili v prostředí grafové databáze Neo4j.

5.1 Důvody vzniku nového schématu

V předchozí kapitole (viz 4) jsme se dozvěděli informace o schématu, který byl vytvořen. Důležité je vědět, proč jsme se rozhodli pro tvorbu nového schématu, když už existuje jiné řešení. Vedly nás k tomu následující důvody:

- Prvním důvodem byla změna modelu MBI (schématu MBI), jehož aktuální verzi můžeme vidět na obrázku 1.1. Hlavní změny se týkaly převážně podpůrných objektů, ke kterým byla přidána hierarchická úroveň. Tým MBI generuje z modelu data, která jsou následovně importována do databáze Neo4j. Tyto změny je nutné reflektovat a vytvořit tak nový návrh.
- Druhý důvod vznikl z potřeby rozšíření schématu. Pro vytvoření dostačujícího základu pro nový dotazovací jazyk musíme vytvořit schéma, které bude optimální.

Rozdíly mezi oběma schématy vyplynou mimo jiné z entit Uzel a Vztah, které obsahují nové atributy, viditelné na obrázku 5.1. U předchozích atributů je jejich význam stejný, liší se akorát název.

5.2 Nové schéma datového úložiště

Vznikající schéma datového úložiště bylo vytvářeno v průběhu realizace této práce. V předchozí kapitole 4 bylo popsáno schéma, ze kterého nové schéma vychází. Návrh procházel postupnými změnami a úpravami.

5.2.0.1 Definování uzlu

Uzly obsahují data objektů MBI. Struktura jednoho uzlu je rozdělena do několika atributů:

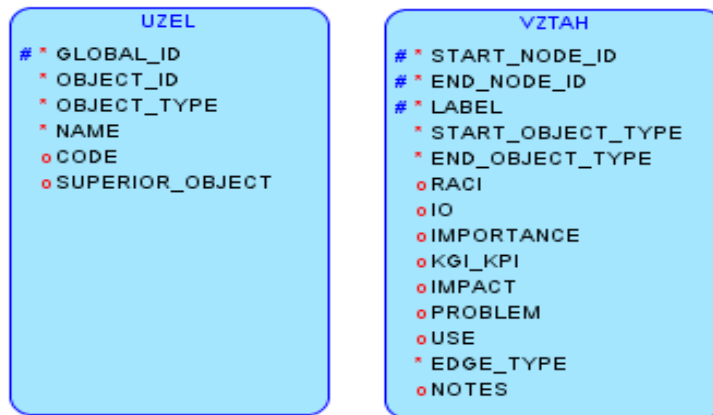
- **GLOBAL_ID** - určuje interní (uměle generovaný) identifikátor uzlu.
- **OBJECT_ID** – identifikátor instance jednoho z objektů MBI mající tvar: [Název objektu]-[Nějaké číslo], například METRIC-86.
- **OBJECT_TYPE** – krátký string (řetězec znaků) popisující slovně objekt, anglický název napsaný velkým písmem. Nabývá právě jedné z třiceti hodnot. Jednou z nich je hodnota MODEL, která identifikuje prvky modelu MBI. Ostatní jsou všechny objekty i se svými hierarchiemi, viditelné na 1.1, například TASK, METRIC, DOMAIN a další.
- **NAME** – slovní popis názvu instance, například „Počty konfiguračních položek“.
- **CODE** – zkratka instance používaná na portálu. Začíná velkým písmenem/písmeny podle typu objektu, následuje několik číslic a může končit znovu velkým písmenem (U001A, UQ002A, R301 a další).
- **SUPERIOR_OBJECT** – nadřazený objekt (jeho OBJECT_ID).

Uzly grafu znovu vyjádříme jako entitu Uzel, kterou si můžeme prohlédnout na obrázku 5.1.

5.2.0.2 Definování hran

Hrany obsahují data, která spojují objekty MBI. Podobně jako uzly obsahují hrany různé atributy:

- **START_NODE_ID** – identifikátor výchozího uzlu (GLOBAL_ID – uměle generovaný – interní).
- **END_NODE_ID** – identifikátor koncového uzlu (GLOBAL_ID – uměle generovaný – interní).



Obrázek 5.1: Nové schéma datového úložiště reprezentované dvěma entitami Uzel a Vztah.

- **LABEL** – popis hrany ve formátu výchozí_koncový uzel, například APPLICATION_METRIC, někdy označován jako typ vazebního objektu. Nabývá právě jedné ze čtrnácti hodnot. Dvanáct z nich je viditelných na 1.1 jako vazby mezi objekty. Další dvě, které jsou nově přidány, jsou INSTANCE_HIERARCHY a INSTANCE_MODEL.
 - **INSTANCE_HIERARCHY** – vyjadřuje vazbu mezi konkrétní instancí objektu a jejím nadřazeným objektem (instancí objektu).
 - **INSTANCE_MODEL** – poukazuje na vazbu mezi instancí a jedním prvkem z modelu MBI (Úloha, Role a další).
- **START_OBJECT_TYPE** – typ objektu výchozího uzlu, například TASK.
- **END_OBJECT_TYPE** – typ objektu koncového uzlu, například ROLE.
- **RACI** - typ vztahu u objektu Role, který dává určité roli vztah k úloze, přesněji kompetence a zodpovědnosti za vykonání úlohy (viz text 1.2.2.4).
- **IO** - Typ vztahu mezi objekty Úloha a Dokument, který určuje, zdali je dokument v úloze: I - vstupní, O - výstupní, U - je v úloze pouze aktualizován.
- **IMPORTANCE** - typ vztahu určující význam mezi objekty. Vyskytuje se u více vazeb, například: TASK_METHOD, TASK_METRIC a další. Nabývá jedné hodnoty z množiny čísel (1,2 a 3) a je použita myšlenka, čím větší číslo, tím větší význam.

- **KGI_KPI** - typ vztahu mezi Úlohou a Metrikou. Identifikátor KGI specifikuje, že se jedná o dosahování cílových hodnot metriky, které jsou významné pro podnik a podnikání. U KPI pak jde o klíčový indikátor výkonnosti dané úlohy.
- **IMPACT** - typ vztahu, který určuje sílu vlivu určitého faktoru na provádění úlohy. Hodnoty mají stejný princip jako u vztahu IMPORTANCE.
- **PROBLEM** - typ vztahu mezi objekty Úloha a Scénář. Určuje čísla dílčích problémů, otázek scénáře, které daná úloha specificky řeší. Má číselné hodnoty. Pokud řeší všechny problémy, je uveden znak „*“.
- **USE** - typ vztahu u Metriky a Dimenze. Znovu nabývá jedné hodnoty z množiny čísel (1, 2 a 3). Tato hodnota určuje četnost využití dimenze při práci s metrikou.
- **EDGE_TYPE** - nový typ, který udává typ vazby. Máme definované tři typy:
 - **II** – Instance-Instance, které reprezentují hrany mezi instancemi na stejné úrovni hierarchie.
 - **IH** – Instance-Hierarchy. Nově vzniklé hrany mezi instancemi a jejich nadřazenými instancemi vyšší hierarchické úrovně.
 - **IM** – Instance-Model. Tento typ vazby je určený pro instance některého z objektů (kterékoliv hierarchické úrovně) a prvek modelu MBI.
- **NOTES** – textové poznámky o vztahu. Tyto poznámky jsou obaleny do značkovacích tagů jazyka HTML, například: `<div><p>Toto je poznámka objektu Úloha.</p></div>`.

Popisy typů vztahů byly čerpány ze zdroje [6].

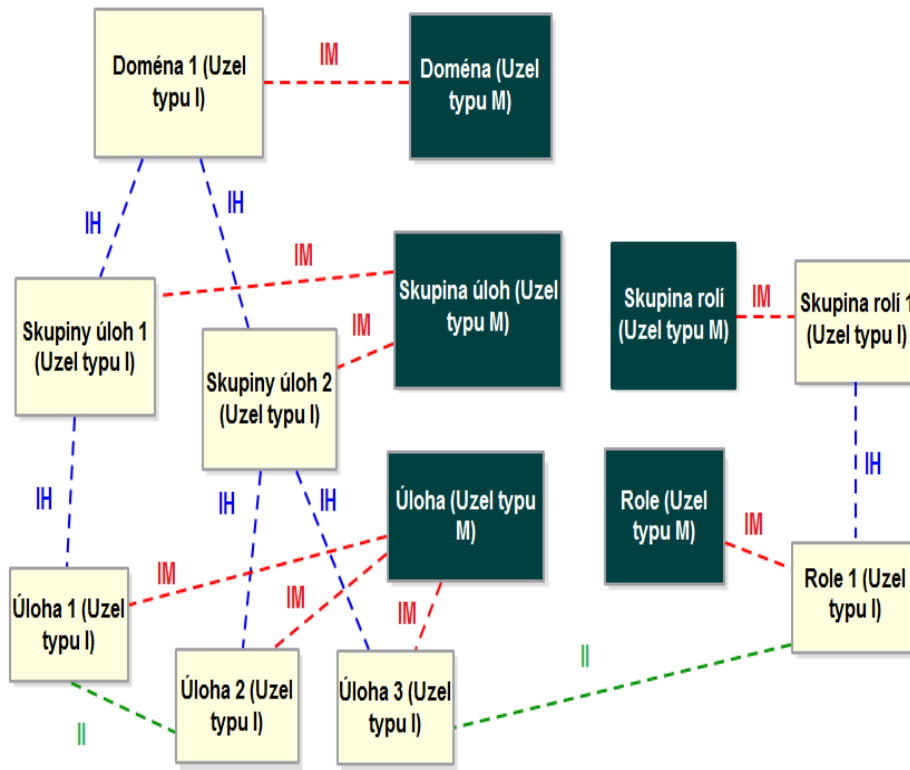
Hrany grafu znovu vyjádříme jako entitu Vztah, kterou si můžeme prohlédnout na obrázku 5.1.

5.2.0.3 Celkový pohled na model

Jestliže si chceme udělat celkový pohled na schéma, můžeme si definovat speciální prvky modelu právě pro tento účel.

Jako první krok můžeme rozdělit entitu Uzel na dva typy:

- **Uzel typu M** – Uzel typu Model, který vyjadřuje prvky modelu MBI (Doména, Role a další). **OBJECT_TYPE** tohoto typu je vždy **MODEL** a váže se na něj Vazba typu **IM** (Instance-Model), viz dále. Jeho další odlišností od druhého typu je ta, že nemá vyplněný atribut **CODE** a **SUPERIOR_OBJECT**. Tyto uzly tvoří kostru modelu MBI.



Obrázek 5.2: Příklad nového schématu datového úložiště.

- **Uzel typu I** – Uzel typu Instance, který reprezentuje jednotlivé instance prvků modelu MBI (Uzlů typu M).

Nyní máme roztříděné Uzly. V dalším kroku roztřídíme Vztahy. Pro tento účel využijeme nového atributu `EDGE_TYPE` u entity `Vztah` a definujeme tak tři vazby schématu datového úložiště:

- **Vazba typu II** – Vazba typu Instance-Instance.
- **Vazba typu IM** – Vazba typu Instance-Model.
- **Vazba typu IH** – Vazba typu Instance-Hierarchy.

Význam těchto vazeb je popsán u atributu `EDGE_TYPE` v části 5.2.0.2.

Částečné grafické vyjádření schématu můžeme vidět v příkladu na obrázku 5.2. Pro větší přehlednost jsou vyobrazeny pouze některé prvky modelu MBI, čili Uzly typu M. Obrázek obsahuje všechny výše definované prvky tohoto schématu.

● METRIC [911]		● TASK_METRIC [46]	
Properties		Properties	
NAME	Počty spravovaných technických prostředků	END_OBJECT_TYPE	METRIC
OBJECT_TYPE	METRIC	START_OBJECT_TYPE	TASK
OBJECT_ID	METRIC-139	EDGE_TYPE	II
GLOBAL_ID	911	NOTES	<div><p>Metrika je orientační vzhledem k hodnocení náročnosti některých operací, jako např. při hodnocení úrovně podnikových procesů, potřebných znalostí zaměstnanců v IT, v manažerských metodách atd.</p></div>
CODE	I601	IMPORTANCE	2
SUPERIOR_OBJECT	METRICSUBGROUP-16	KGI_KPI	KPI

Obrázek 5.3: Ukázka Uzlu METRIC a Vztahu TASK_METRIC.

5.3 Schéma v praxi

Po importu CSV souborů, které obsahují předpřipravená data z XML souboru dodávaného týmem MBI, vzniká nová databáze Neo4j.

Schéma datového úložiště bylo vhodně upravováno, aby mělo rozumnou strukturu na základě požadavků uživatele a vytvořilo vhodný podklad pro nový dotazovací jazyk.

5.3.1 Ukázka modelu

5.3.1.1 Properties

Properties, neboli vlastnosti, neřadíme zcela mezi prvky schématu, nicméně nám o prvcích ukazují důležité informace. Properties se vyskytují ve všech databázích Neo4j u každého uzlu a hrany.

Na snímku 5.3 jsou zobrazeny vlastnosti Uzlu METRIC (Metrika) a Vztahu TASK_METRIC (Úloha_Metrika). Můžeme si všimnout, že obsahují atributy dvou entit z obrázku 5.1 a ověřit si, že je schéma nasazeno správně. Některé atributy (například START_NODE_ID a END_NODE_ID) u Vztahu, v tomto případě Vztah: TASK_METRIC, nejsou zobrazeny, protože hrají pro schéma jinou roli - určují dva uzly, které jsou spojeny tímto Vztahem. Uživatel může zjistit tyto atributy z grafické části zobrazování uzlů a hran, pro příklad viditelné na obrázcích: 5.4 nebo 5.5.

5.3.1.2 Prvky modelu

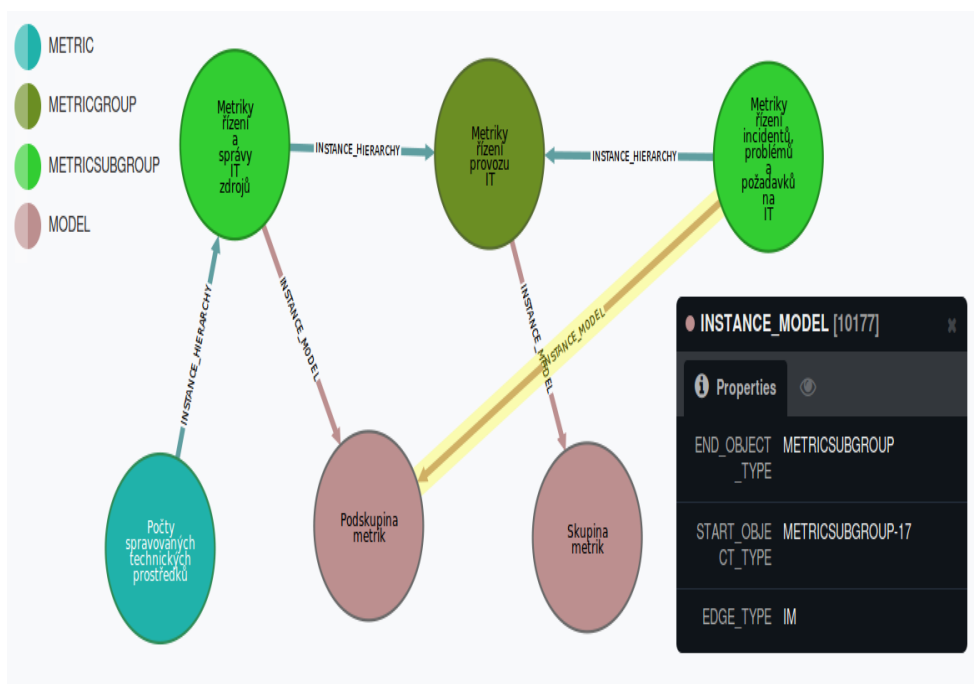
V poslední části této kapitoly si ukážeme samotné prvky schématu, které jsme definovali v předchozím textu. Na demonstraci nám poslouží dva snímky:

- **Hierarchická struktura Metriky, obrázek 5.4** – na tomto obrázku, jak již je podle názvu patrné, vidíme instanci objektu Metrika s názvem: „Počty spravovaných technických prostředků“ a její nadřazenou instanci Podskupiny metrik: „Metriky řízení a správy IT zdrojů“. Tyto dvě instance spojuje Vazba typu Instance-Hierarchy (IH). V pravé části obrázku jsou však vyobrazeny vlastnosti jiné vazby, Vazby typu Instance-Model (IM). Viditelné jsou další uzly a hrany, které není třeba dále popisovat.

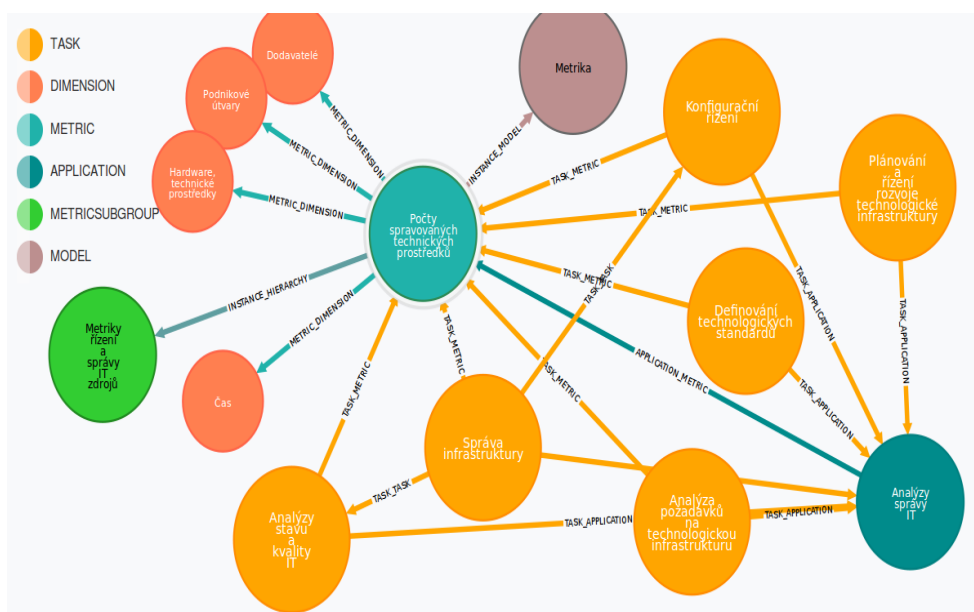
Pro zajímavost by bylo vhodné zmínit, proč není přítomen uzel s názvem: „Metrika“ a s ním spojená vazba k jediné instanci objektu Metrika. Při tvorbě tohoto snímku byl jako úvodní uzel pro procházení databáze pomocí Cypher dotazu vybrán uzel s názvem: „Metriky řízení provozu IT“. Po dvou krocích průchodu databáze způsobem „poklikání“ na uzel, viz také text 3.1.3.1, se došlo k tomuto výsledku. Při třetím kroku „poklikání“ na „Počty spravovaných technických prostředků“, při kterém by se ve výsledku objevil zmiňovaný Uzel typu M (Metrika), došlo k výskytu většího množství uzlů a obrázek by byl pro čtenáře nepřehledný. Byl tak záměrně vytvořen druhý snímek, který si nyní představíme.

- **Okolí instance: „Počty spravovaných technických prostředků“, obrázek 5.5** – jediná vazba, která není přítomna na předchozím obrázku 5.4, je Vazba typu Instance-Instance (II). Hlavně z tohoto důvodu byl jako názorný příklad vybrán tento snímek, ve kterém se tento typ nachází. Druhý důvod je popsán výše. Na první pohled možná někteří uživatelé v úvodu nepoznají, že se zde vyskytují Vazby typu II, jelikož je u hran vidět atribut LABEL, nikoliv EDGE_TYPE. Uživatelé později poznají, že se atribut LABEL a EDGE_TYPE u Vazeb typu IH a IM v podstatě neliší a má stejný význam. V jednom případě je akorát zapsán zkratkou, v druhém celým názvem. Vazby typu II jsou tedy všechny ostatní vazby a jsou přítomné mezi instancemi objektů MBI (nejnižší hierarchické úrovně), na tomto snímku například: TASK_TASK, TASK_METRIC a další.

5. NÁVRH NOVÉHO SCHÉMATU DATOVÉHO ÚLOŽIŠTĚ



Obrázek 5.4: Ukázka části modelu - hierarchická struktura Metriky.



Obrázek 5.5: Ukázka části modelu - okolí instance: „Počty spravovaných technických prostředků“.

Návrh dotazovacího jazyka

Cílem této kapitoly je návrh graficky orientovaného a doménově specifického dotazovacího jazyka, který bude následně překládán/transformován do jazyka Cypher. Pomocí jazyka Cypher bude probíhat dotazování na grafovou databázi Neo4j.

V této kapitole si nový jazyk popíšeme. V první části se zaměříme na zvolený postup návrhu a jazyk si také definujeme. Navazující text se věnuje samotnému jazyku, který je záměrně rozdělen na dvě části. V poslední sekci se dozvíme informace o metodě nazvané „Asistované procházení grafem“, která s jazykem souvisí.

6.1 Průběh návrhu

V poslední části druhé kapitoly (viz 2.4) jsme uvedli výsledek prováděné analýzy, ze které budeme při návrhu vycházet. Na základě uživatelských potřeb jsme se rozhodli navrhnout jazyk způsobem, který by zohledňoval dotazování se na konkrétní objekt a v druhé řadě na vazby mezi těmito objekty. Druhý způsob by pak umožňoval uživateli procházení modelem již při tvorbě dotazu a získávání souvislostí mezi objekty. Dotazovací jazyk by měl navíc být pro uživatele intuitivní.

Dalším aspektem, který je potřeba připomenout, je jazyk Cypher. Ten jsme zařadili mezi cílové technologie a byl rozebrán v podkapitole 3.2. Jazyk Cypher se používá při práci s grafovou databází Neo4j, která je datovým úložištěm nového schématu (grafu), popsáném v předchozí kapitole (viz 5), a obsahuje tak data z MBI, na která se chce uživatel ptát.

Zvolené řešení je vytvoření dotazovacího jazyka, který bude rozdělen na dvě části: „Specificky orientované dotazování“ a „Dotazování pomocí vazeb“. Tyto dotazovací způsoby budou používat převážně grafické prvky, ze kterých uživatel postaví svůj dotaz. Na základě zvolených prvků a jejich specifikaci bude vytvořen dotaz, který se převede/přeloží do jazyka Cypher.

Pro návrh jazyka byly vytvořeny vzorové dotazy, které by měly co nejvíce vyhovovat požadavkům uživatele.

Ještě před tím, než začneme nový jazyk popisovat, si ho definujeme.

6.1.1 Definice jazyka

MBIQL (Management of Business Informatics Query Language) je graficky orientovaný a doménově specifický dotazovací jazyk, který je určen pro dotazování se na data MBI. Dotaz jazyka MBIQL je vytvořen pomocí jedné ze dvou metod: Specificky orientované dotazování a Dotazování pomocí vazeb. Výsledný dotaz je následně automaticky překládán do jazyka Cypher.

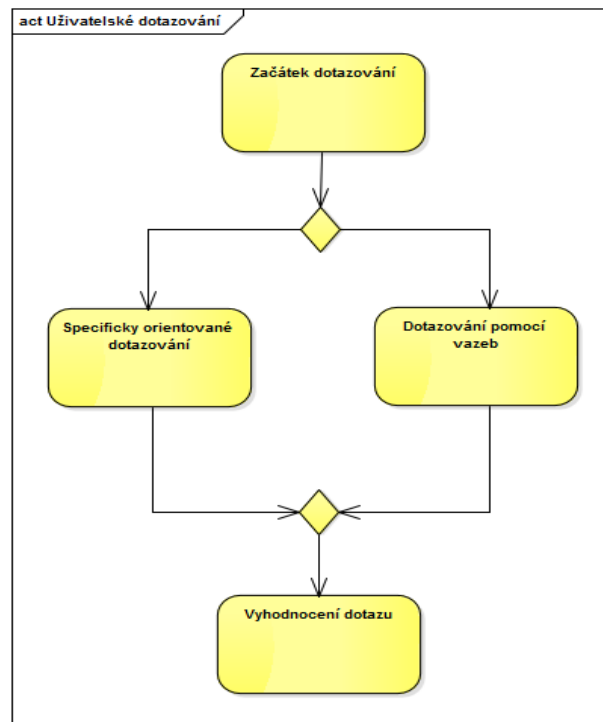
6.1.2 Vzorové dotazy

Vzorové dotazy byly použity převážně u způsobu Dotazování pomocí vazeb. Pro Specificky orientované dotazování byl využit pouze jeden dotaz:

1. *Zobraz instanci objektu Úloha, která má název: „Správa infrastruktury“.*

Bližší rozebrání dotazu a návrhu Specificky orientovaného dotazování naleznete v podkapitole 6.2. Informace o metodě Dotazování pomocí vazeb získáte v sekci 6.3. Seznam dotazů pro Dotazování pomocí vazeb:

1. *Zobraz všechny Metody, které mají vztah k Úloze: „Propojení metrik byznysu a metrik IT“.*
2. *Zobraz všechny Skupiny úloh, které spadají do Domény: „Strategické řízení IT“.*
3. *Najdi pouze Faktory, které jsou využity u Scénáře: „Je otázka, jak se má CIO připravit na poradu vedení podniku“.*
4. *Zjistí, do jaké Skupiny metrik patří Metrika: „Počty spravovaných technických prostředků“. Zobraz všechny hrany a uzly.*
5. *Zjistí, do jaké Skupiny metrik patří Metrika: „Počty spravovaných technických prostředků“. Zobraz všechny hrany a uzly. Zobraz také uzly prvků modelu MBI (Uzly typu M).*
6. *Zobraz kompletní hierarchii dvou Rolí, první s názvem: „Návrhář databázi“, druhou s kódem: „R108“. K nim zobraz také všechny Úlohy, se kterými jsou vázány.*



Obrázek 6.1: Životní cyklus dotazu.

6.1.3 Rozdělení jazyka

V předchozích částech textu této kapitoly jsme se dozvěděli, že nový jazyk bude rozdělen na dvě základní části (metody), které při dotazování určí, jakým směrem se uživatel vydá.

Pro grafické znázornění průběhu dotazu bylo vytvořeno několik schémat, které si představíme. První z nich můžeme vidět na obrázku 6.1, na kterém je zobrazen životní cyklus dotazu. Po začátku dotazování si uživatel musí vybrat jednu z metod. Po jejím ukončení dojde k vyhodnocení dotazu a s ním spojené ukončení dotazování.

6.1.4 Využití grafických prvků

Použití grafických prvků při dotazování (graficky orientované jazyky) je intuitivnější než textově orientované dotazovací jazyky, které tyto prvky nevyužívají. Uživatelé se snadněji naučí dotazování používat, dokáží podle grafických prvků v mnohých případech předvídat jejich chování pro jazyk a nemusí se učit nové příkazy textově orientovaného dotazovacího jazyka.

Pro nový dotazovací jazyk jsou v obou jeho metodách použity grafické prvky. Mezi základní prvky zařadíme „Objekt“ a „Vazbu“.

- **Objekt** – je využit u obou způsobů dotazování. Jeho role je stejná jako v modelu MBI. Objekt totiž představuje všechny neurčité instance jednoho konkrétního objektu z MBI (i jeho hierarchií). Při dotazování jsou zpřístupněny všechny Objekty, viditelné na obrázku 1.1.
- **Vazba** – je využita pouze u Dotazování pomocí vazeb. Její role je znovu podobná jako v modelu MBI. Jedinou hlavní odlišností je, že má směr. Jedna Vazba spojuje právě dva Objekty. Podle výchozího a koncového Objektu poznáme, jaký směr Vazba má. U Objektů, které popisují úrovně hierarchií, je vždy následující směr:

Objekt -> Podskupina -> Skupina.

Do ostatních grafických prvků patří „Seznam“, „Pole příznaků“, „Pole logických operátorů“ a „Textové pole“.

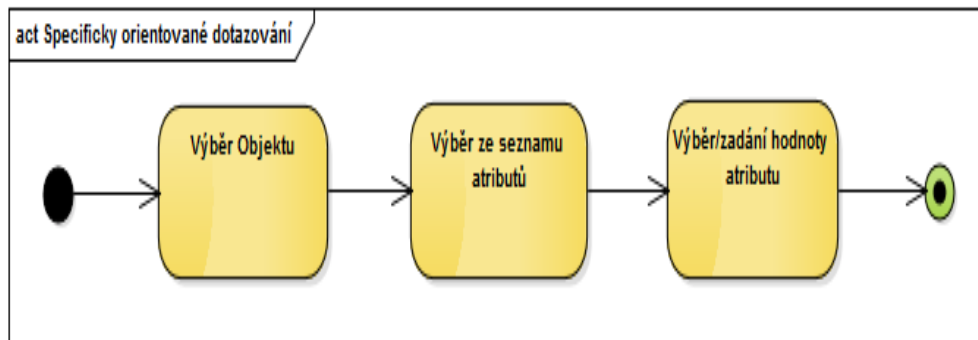
- **Seznam** – používá se k vybírání hodnot atributů entit Uzel a Vztah, viditelné na obrázku 5.1.
- **Pole příznaků** – použito pouze u Dotazování pomocí vazeb ve třech krocích: Rozhodnutí o prvcích modelu MBI (Uzlů typu M), Výběr prvků ze Struktury dotazu a ve Výběru návratových hodnot.
- **Pole logických operátorů** – místo, kde se sestavuje pořadí logických operátorů mezi prvky Objekt a Vazba nebo mezi atributy těchto prvků (viz. text 6.3.1.2).
- **Textové pole** – uživatelé zadávají do textového pole hodnoty atributů. Alternativa pro Seznam. Při zadávání je poskytnuta pomoc našeptávače.

Využití výše uvedených grafických prvků bude upřesňováno v dalších částech kapitoly.

6.2 Specificky orientované dotazování

Prvním dotazovacím způsobem je Specificky orientované dotazování, které se zaměřuje na konkrétní instance objektů. Vzorovým příkladem je, že má uživatel potřebu najít instanci, která je specifikovaná nějakým atributem, který buďto dopředu zná, nebo si ho chce vyhledat při samotném dotazování.

Po ukončení dotazování a jeho vyhodnocení bude uživateli umožněno zobrazit výslednou instanci. Z ní pak chce čerpat konkrétní data, která se může dozvědět. Jednou z navrhovaných možností, jak s instancí dále pracovat, je použití vlastnosti procházení grafové databáze Neo4j, popsané v textu 3.1.3.1. Specificky orientovaným dotazováním jsme našli začínající uzel, ze kterého můžeme prozkoumávat okolí uzlu do všech směrů. Tato vlastnost Neo4j nicméně nesouvisí s dotazovacími způsoby.



Obrázek 6.2: Metoda Specificky orientované dotazování.

6.2.1 Průběh metody

Na obrázku 6.2 je vyjádřen průběh metody Specificky orientovaného dotazování.

Prvním krokem, který je nutné po zvolení této metody udělat, je vybrání objektu, ke kterému se vztahuje hledaná instance. Dále musíme vybrat jeden atribut ze seznamu atributů daného objektu. V posledním kroku následuje zadání hodnoty. Uživatel může vybírat ze Seznamu, kde jsou zobrazeny všechny dostupné hodnoty pro daný atribut přítomné v datovém úložišti k danému objektu, nebo zadat hodnotu do Textového pole, kde se mu nabízí pomoc našeptávače.

6.2.1.1 Ukázka konkrétního dotazu

Jako ukázkový dotaz, na kterém si odsimulujeme průběh dotazování pomocí metody Specificky orientovaného dotazování, byl zvolen dotaz:

1. *Zobraz instanci objektu Úloha, která má název: „Správa infrastruktury“.*

V prvním kroku musíme zvolit konkrétní Objekt z grafických prvků typu Objekt. Vybereme tedy Úlohu. V dalším kroku vybereme atribut, který nám nabízí entita Uzel (jelikož jsme vybrali Objekt), viditelná na obrázku 5.1, pro OBJECT_TYPE = TASK. Seznam atributů entity Uzel, kromě již specifikovaného atributu OBJECT_TYPE, se zobrazí v grafickém prvku Seznam. Pro uživatele se mohou zobrazovat různé aliasy na základě implementace jazyka, čili atribut NAME může být uváděn ve výčtu například jako Název. V poslední části vybereme hodnotu ze seznamu aktuálních instancí dostupných v datovém úložišti Neo4j. Tento seznam instancí se nám zobrazí znovu v Seznamu. Druhou možností, jak vybrat hodnotu, je její zadání do Textového pole, v tomto případě hodnotu „Správa infrastruktury“.

Po provedených krocích dojde k vyhodnocení dotazu, kdy je dotaz přepsán do jazyka Cypher.

6.2.1.2 Transformace dotazu do jazyka Cypher

U Specificky orientovaného dotazování je závěrečná transformace do jazyka Cypher relativně triviální. Můžeme použít šablonu, ze které budeme u každého dotazu vycházet:

```
MATCH (a. [OBJECT_TYPE])
WHERE a. [ATRIBUT] = '[HODNOTA]'
RETURN a;
```

Pro vzorový dotaz rozebraný v předchozí části textu (viz 6.2.1.1), má výsledný Cypher dotaz tuto podobu:

```
MATCH (a.TASK)
WHERE a.NAME = 'Správa infrastruktury'
RETURN a;
```

V průběhu dotazování se ukládají vybrané prvky dotazu a při transformaci jsou doplněny do uváděné šablony.

6.3 Dotazování pomocí vazeb

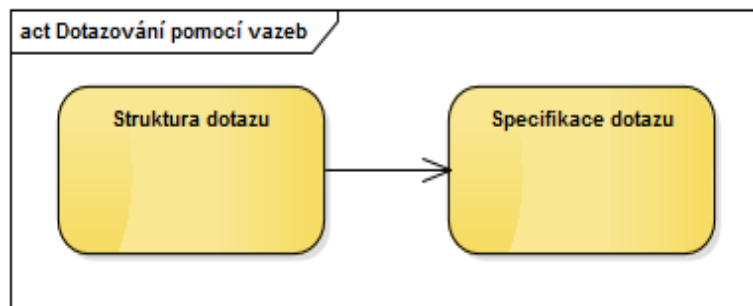
Druhou metodou, která je součástí jazyka MBIQL, je Dotazování pomocí vazeb. Na rozdíl od Specificky orientovaného dotazování se Dotazování pomocí vazeb zaměřuje na vztahy mezi objekty MBI a hledání souvislostí.

6.3.1 Rozdělení a průběh metody

Metoda je rozdělena na dvě části:

- **Struktura dotazu** – ve Struktuře dotazu, jak již je z názvu patrné, půjde o vytvoření kostry dotazu. Konkrétně se jedná o vybrání grafických prvků Objekt a Vazba.
- **Specifikace dotazu** – nad vybranými prvky ze Struktury dotazu proběhne jejich specifikace. Po této části je dotaz kompletní.

Grafické vyjádření průběhu metody Dotazování pomocí vazeb můžeme vidět na obrázku 6.3.



Obrázek 6.3: Metoda Dotazování pomocí vazeb.

6.3.1.1 Průběh Struktury dotazu

Prvním úkolem pro uživatele je zvolit Objekt, od kterého se začne „pohybovat“ - počáteční Objekt. Uživatel může vybírat ze všech dostupných Objektů, které mají Vazby. Jelikož z každého Objektu vede alespoň jedna Vazba (všechny objekty mají totiž minimálně dvě hierarchické úrovně), může uživatel vybírat ze všech. V následujících cyklech již musí zvolit Vazby. Uživatel má možnost vybírat z těchto Vazeb:

- Vazby, které vedou z/do počátečního Objektu.
- Vazby, které vedou z/do Objektu, na který/ze kterého vedla Vazba vybraná v některém z předchozích průchodů cyklem.

Cyklus se opakuje, dokud uživatel nezvolí všechny požadované Vazby, které chce použít při dotazování. Každá Vazba může být vybrána jen jednou. Konkrétní příklad bude uveden na vzorových dotazech v části textu 6.3.2.2 a také v části 8.3.1, ve které je uveden vhodný obrázek 8.4 pro grafickou představu výběru počátečního Objektu a Vazeb.

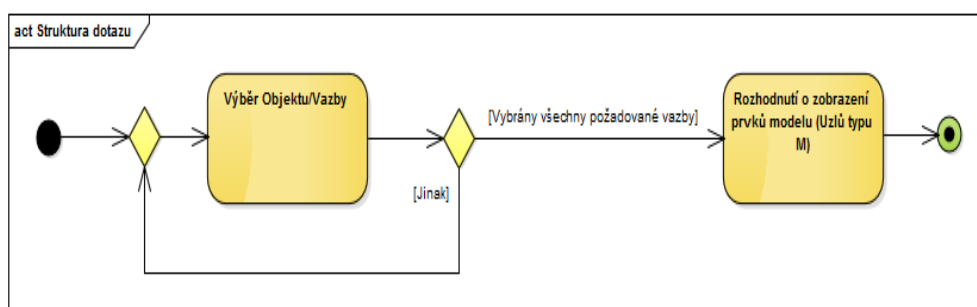
Ještě před ukončením Struktury dotazu musí uživatel učinit rozhodnutí o zobrazení prvků modelu (Uzlů typu M) k vybraným Objektům. Pokud je bude chtít zobrazit, ve výsledku se k množině uzlů zobrazí i příslušné Uzly typu M, viz obrázek 5.2.

Vybranými prvky ze Struktury dotazu se předpřipraví i struktura pro Cypher dotaz, konkrétně pattern (vzor), který se bude následně hledat ve schématu.

Průběh Struktury dotazu si můžeme prohlédnout na obrázku 6.4.

6.3.1.2 Průběh Specifikace dotazu

Ve Specifikaci dotazu se na jejím začátku pracuje s prvky (Objekt či Vazba), které byly vybrány ve Struktuře dotazu. Uživatel zaškrtně v Příznakovém poli prvky, které chce nějakým způsobem specifikovat. Jestliže se jedná o více



Obrázek 6.4: Průběh části Struktura dotazu.

prvků, vybere uživatel z logických operátorů OR nebo AND, které mají být mezi jednotlivými prvky a také jejich uspořádání včetně uzávorkování (např. PrvekA OR (PrvekB AND PrvekC)). Uživatel může pořadí prvků, operátorů a závorek volně měnit v Poli logických operátorů.

Následuje smyčka, ve které uživatel zvolí atributy na prvním prvku. U něj napíše hodnotu do Textového pole nebo ji vybere ze Seznamu, podobně jako v případě Specificky orientovaného dotazování. Jestliže se uživatel rozhodne omezit prvek upřesněním dalšího atributu, opakuje se výběr atributu. Uživatel si znovu vybírá ze dvou logických operátorů, které budou použity mezi atributy, jako v případě prvků z prvního kroku této metody. Výsledné uspořádání v Poli logických operátorů u prvků a atributů se projeví po transformaci na Cypher dotaz v klauzuli WHERE. Celý cyklus se opakuje do té doby, dokud nebyly tímto způsobem specifikovány všechny vybrané prvky z prvního kroku v části Specifikace dotazu.

Posledním krokem, který je potřeba učinit před ukončením metody Dotazování pomocí vazeb, je určit návratové hodnoty dotazu. Výběr se znovu týká pouze prvků zvolených ve Struktuře dotazu a probíhá přes zaškrťávání v Příznakovém poli.

Průběh Specifikace dotazu můžeme vidět na obrázku 6.5.

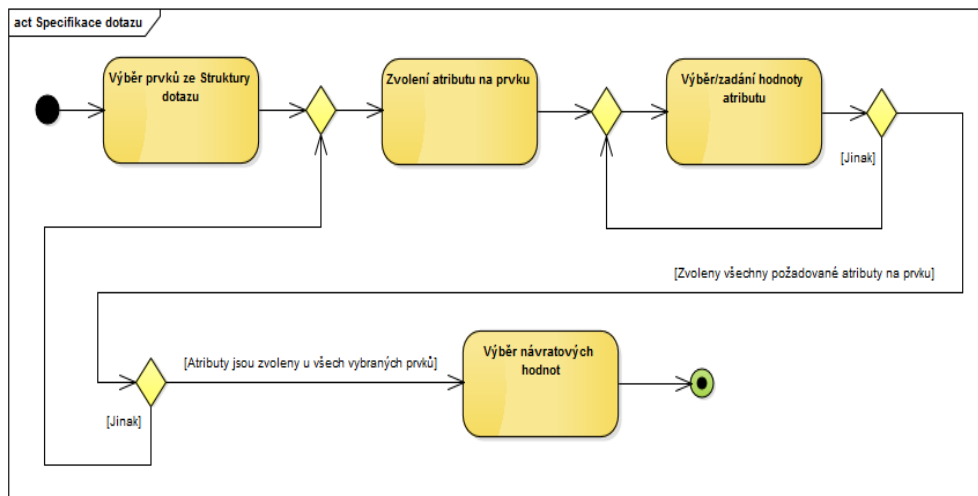
6.3.2 Ukázky dotazů a jejich transformace

V následující části textu si projdeme vybrané dotazy ze vzorových dotazů uvedených v textu 6.1.2. U dotazů si popíšeme jejich vytváření v průběhu metody a nastíníme jejich transformaci/překlad do jazyka Cypher.

6.3.2.1 Dotazy na přímé vazby

První dva dotazy se zaměřují na použití přímých vazeb:

1. *Zobraz všechny Metody, které mají vztah k Úloze: „Propojení metrik byznysu a metrik IT“.*



Obrázek 6.5: Průběh části Specifikace dotazu.

Nejprve musíme zvolit Objekt, od kterého budeme začínat. Vybereme tedy Objekt Úloha, dotaz by fungoval i kdybychom vybrali jako první Objekt Metoda, byl by jen jinak „poskládaný“. V následujícím cyklu vybereme Vazbu TASK_METHOD, která je graficky vyjádřena mezi Objekty Úloha a Metrika, pro upřesnění směru: Úloha -> Metrika. Podle zadání dotazu vidíme, že už máme vybrány všechny prvky ze Struktury dotazu. V tomto případě navíc nechceme zobrazovat Uzly typu M, Struktura dotazu proto končí a startuje Specifikace dotazu.

Vzor pro Cypher dotaz:

```
(a:TASK)-[rel:TASK_METHOD]->(b:METHOD)
```

Ve Specifikaci dotazu vybereme v prvním kroku pouze Objekt Úloha. U Úlohy zvolíme atribut Název (NAME), u kterého nastavíme hodnotu: „Propojení metrik byznysu a metrik IT“ jedním ze způsobů, které metoda dovoluje (výběr ze Seznamu, zadání hodnoty do Textového pole). Žádný další atribut omezovat nechceme, všechny prvky byly specifikovány, vybereme tedy návratové hodnoty: Objekt (Metoda, Úloha), Vazba (Úloha -> Metoda). Rozhodli jsme se zobrazit všechny prvky, nejen Objekt Metoda, jak by ze zadání mohlo vyplývat.

Výsledný Cypher dotaz:

```
MATCH (a:TASK)-[rel:TASK_METHOD]->(b:METHOD)
WHERE a.NAME = 'Propojení metrik byznysu a metrik IT'
RETURN a, rel, b;
```

2. *Zobraz všechny Skupiny úloh, které spadají do Domény: „Strategické řízení IT“.*

Průběh dotazu je stejný jako u předchozího, liší se akorát vybrané prvky a hodnoty.

Výsledný Cypher dotaz:

```
MATCH (a:TASKSGROUP)-[rel:INSTANCE_HIERARCHY]->
(b:DOMAIN)
WHERE b.NAME = 'Strategické řízení IT'
RETURN a, rel, b;
```

Na příkladu můžeme vidět, že Objekty hierarchické úrovně mají opravdu Vazby, které mají směr Podskupina -> Skupina, jsou navíc vždy typu INSTANCE_HIERARCHY. Cypher dotaz by v současné době vrátil správný výsledek i bez určení LABEL (INSTANCE_HIERARCHY) u Vazby rel, jelikož jsou výběrem Objektů nastaveny uzly na TASKSGROUP a DOMAIN. Jiná Vazba tuto dvojici aktuálně nespojuje.

6.3.2.2 Dotazy na více vazeb

V Dotazování pomocí vazeb je uživateli povoleno využívat více řetězených vazeb.

3. *Najdi pouze Faktory, které jsou využity u Scénáře: „Je otázka, jak se má CIO připravit na poradu vedení podniku“.*

Požadujeme zobrazit pouze Faktory, jako první Objekt vybereme tedy Faktor. Znovu by ale mohl být vybraný jiný Objekt z výsledné trojice Objektů. Podle předchozích zkušeností a znalostí, nebo podle obrázku modelu MBI (viz 1.1), usoudíme, že v dalším kroku vybereme Vazbu (Faktor <- Úloha) a v kroku následujícím pak Vazbu (Úloha -> Scénář), kdy dostaneme hledané spojení. Máme vybráno celkem pět prvků: Objekty (Faktor, Úloha, Scénář) a zmiňované Vazby mezi nimi.

Vzor pro Cypher dotaz:

```
MATCH (a:FACTOR)<-[relTaskFactor:TASK_FACTOR]-(b:TASK),
(b:TASK)-[relTaskScenario:TASK_SCENARIO]->(c:SCENARIO)
```

Specifikace dotazu proběhne vybráním jediného prvku: Objekt (Scénář) a jediného atributu: Název (NAME), u kterého zadáme jedním z možných způsobů hodnotu: „Je otázka, jak se má CIO připravit na poradu vedení podniku“. Jako návratová hodnota jsou vybrány pouze uzly Faktoru (prvek Objekt: Faktor).

Výsledný Cypher dotaz:

```
MATCH (a:FACTOR)-[relTaskFactor:TASK_FACTOR]-(b:TASK),
      (b:TASK)-[relTaskScenario:TASK_SCENARIO]->(c:SCENARIO)
WHERE c.NAME = 'Je otázka, jak se má CIO připravit na
poradu vedení podniku'
RETURN a;
```

4. *Zjistí, do jaké Skupiny metrik patří Metrika: „Počty spravovaných technických prostředků“. Zobraz všechny hrany a uzly.*

Průběh dotazu je stejný jako u předchozího, liší se akorát vybrané prvky a hodnoty (i návratové).

Výsledný Cypher dotaz:

```
MATCH (a)-[relMetricsubgroup:INSTANCE_HIERARCHY]->(b),
      (b)-[relMetricgroup:INSTANCE_HIERARCHY]->(c)
WHERE a.NAME = 'Počty spravovaných technických
prostředků'
RETURN a, relMetricsubgroup, b, relMetricgroup, c;
```

Z Cypher dotazu vidíme, že pro správný výsledek je možné vynechat OBJECT_TYPE u prvků Objekt. U dotazu číslo 2 jsme naopak zjistili, že je možné vynechat LABEL u Vazby. Neměli bychom ale vynechávat oba atributy. Pro jistotu, že nám dotaz vrátí správná data, bude lepší při transformaci udávat oba tyto atributy, jejichž hodnoty poznáme z vybraných grafických prvků (Objekt a Vazba).

6.3.2.3 Dotaz na Uzly typu M

Pro přidání Uzlů typu M do dotazu byl účelně upraven 4. dotaz do této podoby:

5. *Zjistí, do jaké Skupiny metrik patří Metrika: „Počty spravovaných technických prostředků“. Zobraz všechny hrany a uzly. Zobraz také uzly prvků modelu MBI (Uzly typu M).*

Jediná odlišnost tohoto dotazu s dotazem číslo 4 je v posledním kroku Struktury dotazu, kdy se rozhodujeme o přidání Uzlů typu M. Jelikož máme vybrány tři Objekty (Faktor, Úloha, Scénář), přidají se k nim další tři nové Objekty, které nazveme Objekty typu M a reprezentují Uzly typu M. K těmto novým Objektům se přidají i nové Vazby - k dosavadním dvěma z předešlého dotazu přidáme tři nové. Celkově máme šest Objektů a pět Vazeb. Nové Vazby jsou typu INSTANCE_MODEL, jelikož spojují Uzly typu I a Uzly typu M (viz obrázek 5.2).

Výsledný Cypher dotaz:

```
MATCH (a)-[rel1:INSTANCE_HIERARCHY]->(b),
```

```
(a)-[relModel:INSTANCE_MODEL]->(d),
(b)-[rel2:INSTANCE_HIERARCHY]->(c),
(b)-[relModel2:INSTANCE_MODEL]->(e),
(c)-[relModel3:INSTANCE_MODEL]->(f)
WHERE a.NAME = 'Počty spravovaných technických
prostředků'
RETURN a, rel1, relModel1, b, rel2, relModel2,
c, relModel3, d, e, f;
```

6.3.2.4 Dotaz na omezení více atributů

Poslední typový dotaz, který jsme ještě nepředstavili, je dotaz na omezení více atributů jednoho prvku, v našem případě Objektu Role.

1. *Zobraz kompletní hierarchii dvou Rolí, první s názvem: „Návrhář databázi“, druhou s kódem: „R108“. K nim zobraz také všechny Úlohy, se kterými jsou vázány.*

V dotazu začneme od Úlohy, kterou požadujeme zobrazit. Průběh metodou dotazu je znovu podobný jako v předchozích případech - spojíme: Úloha -> Role a Role -> Skupina rolí. U Specifikace dotazu vybereme Objekt Role a v nadcházejícím cyklu omezíme Objekt u atributu Název (NAME) hodnotou 'Návrhář databázi' a u atributu Kód (CODE) hodnotou „R108“. Zvolené atributy budou po transformaci do Cypher dotazu odděleny logickým operátorem OR, který jsme specifikovali v Poli logických operátorů: Název OR Kód.

Výsledný Cypher dotaz:

```
MATCH (a)-[relRole:TASK_ROLE]->(b),
(b)-[relRoleHierarchy:INSTANCE_HIERARCHY]->(c)
WHERE b.NAME = 'Návrhář databázi' OR b.CODE = 'R108'
RETURN a, relRole, b, relRoleHierarchy, c;
```

6.4 Asistované procházení grafem

Po úspěšném vytvoření dotazu a jeho následné transformaci na dotaz z jazyka Cypher můžeme jednoduchým způsobem získávat data MBI z prostředí grafové databáze Neo4j, která je datovým úložištěm nového schématu. Cypher dotaz vrací data v podobě grafu (uzlů a hran). Dotazem, který vznikl metodou Dotazování pomocí vazeb, popřípadě variantou procházení okolí (netýká se jazyka MBIQL), kterou umožňuje Neo4j, může být uživateli zobrazen graf, ze kterého chce vidět jen určitou část. Pro takový případ navrhujeme metodu „Asistované procházení grafem“. U této metody bude nastíněna jen její částečná funkcionálnost.

6.4.1 Zobrazování pomocí vazeb

Pro návrh Asistovaného procházení grafem můžeme využít Typy vazeb, které jsme definovali v textu 5.2.0.3. Pro připomenutí máme ve schématu tři Typy vazeb: IM (Instance-Model), IH (Instance-Hierarchy) a II (Instance-Instance).

Pro uživatele by mohlo být zajímavé, kdybychom zobrazovali pouze uzly, které spojuje jeden z vybraných Typů vazeb. Definujeme si proto tři nové metody „Procházení grafu“:

- **Procházení IM** - Procházení Instance-Model, které projde celý graf a zobrazí pouze uzly a hrany, které mají atribut `EDGE_TYPE = IM`.
- **Procházení IH** - Procházení Instance-Hierarchy, které projde celý graf a zobrazí pouze uzly a hrany, které mají atribut `EDGE_TYPE = IH`.
- **Procházení II** - Procházení Instance-Hierarchy, které projde celý graf a zobrazí pouze uzly a hrany, které mají atribut `EDGE_TYPE = II`.

6.4.2 Zobrazování pomocí jiných prvků

S grafem bychom mohli pracovat i pomocí jiných prvků. Kdyby si uživatel například rozmyslel, jaké Objekty (`OBJECT_TYPE` u entity Uzel) nechce vidět - u Dotazování pomocí vazeb si návratovou hodnotu uživatel volí v posledním kroku, mělo by mu být umožněno tyto Objekty z grafu odebrat.

Metody na odebrání uzlů by měly být povoleny pro všechny uzly ze schématu s odlišnými hodnotami atributu `OBJECT_TYPE` (`TASK`, `METRIC`, `MODEL` atd.), pokud se v grafu nacházejí.

6.4.3 Asistované procházení

Poslední navrhovanou částí, kterou si uvedeme, bude „Asistované procházení“. Jeho fungování by mohlo být založeno na principu, ve kterém by si uživatel zvolil výchozí uzel. Následně by mu bylo nabídnuto, po jaké hraně se chce vydat:

- Pouze po jedné hraně s možností omezení atributu na hraně (entita Vztah) nebo s možností specifikovat cílový uzel.
- Po více hranách se stejnými možnostmi jako v předchozím případě.
- Po všech hranách bez specifikace atributu.

Po kroku (iteraci), ve kterém si uživatel vybral „cestu po hranách“, by se ostatní hrany, které vedou z výchozího uzlu z grafu a nevyhovují označení či specifikaci, mohly odebrat. Mohly by se také odstranit uzly, které jsou po odebrání hran v grafu bez jakékoliv hrany.

6. NÁVRH DOTAZOVACÍHO JAZYKA

Tento krok by mohl uživatel opakovat z uzlů, na které vedly hrany z předchozí iterace. Uživatel by mohl Asistované procházení v libovolný moment ukončit.

Porovnání jazyka s jiným

V této kapitole se budeme zabývat krátkým porovnáním nového jazyka MBIQL s jiným doménově specifickým dotazovacím jazykem. Jelikož nově navržený jazyk obsahuje grafické prvky, pokusíme se na ně zaměřit i u hledaného jazyka. Nejprve si ho však představíme.

7.1 Grafický dotazovací jazyk GQL

Pro průzkumu doménově specifických jazyků byl vybrán jako nejvhodnější jazyk pro porovnání grafický dotazovací jazyk GQL. Všechny informace týkající se tohoto jazyka jsou čerpány ze zdroje [3], kromě informací ze zdrojů uváděných v textu.

7.1.1 Seznámení s jazykem

Grafický dotazovací jazyk GQL ⁷ byl navržen trojicí autorů z Lotyšské univerzity. Důvodem vzniku byla myšlenka vytvoření grafického jazyka, který umožní jeho uživatelům získávat strukturovaná data.

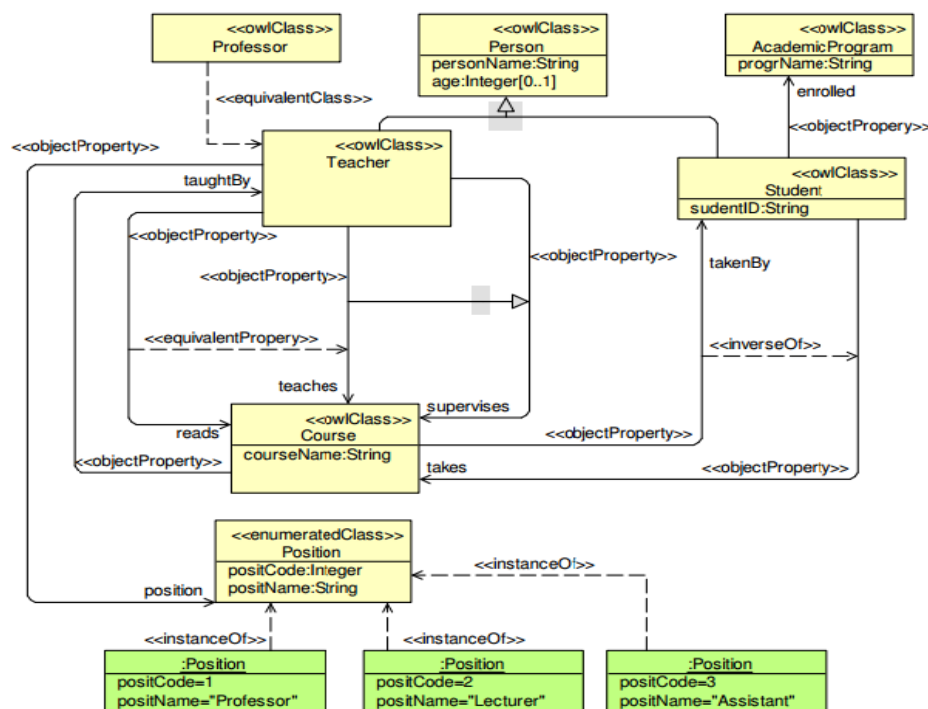
Jazyk GQL je založený na rysech tří jazyků:

- **OWL** - ontologický jazyk OWL ⁸, který slouží k popisu tříd a relací a měl by být základním jazykem sémantického webu. Jazyk OWL také dokáže vytvořit komplexní vztahy i mezi třídami a vlastnostmi [18]. Pro Grafický dotazovací jazyk GQL byla použita varianta jazyka OWL, konkrétně OWL DL.
- **UML** - grafický jazyk UML ⁹ se používá pro vizualizaci, specifikaci, konstrukci a modelování při objektově orientované analýze [19]. Na jazyku UML je založena grafická stránka jazyka GQL.

⁷GQL - Graphical Query Language.

⁸OWL - Web Ontology Language.

⁹UML - Unified Modeling Language.



Obrázek 7.1: Schématická část ontologie [3].

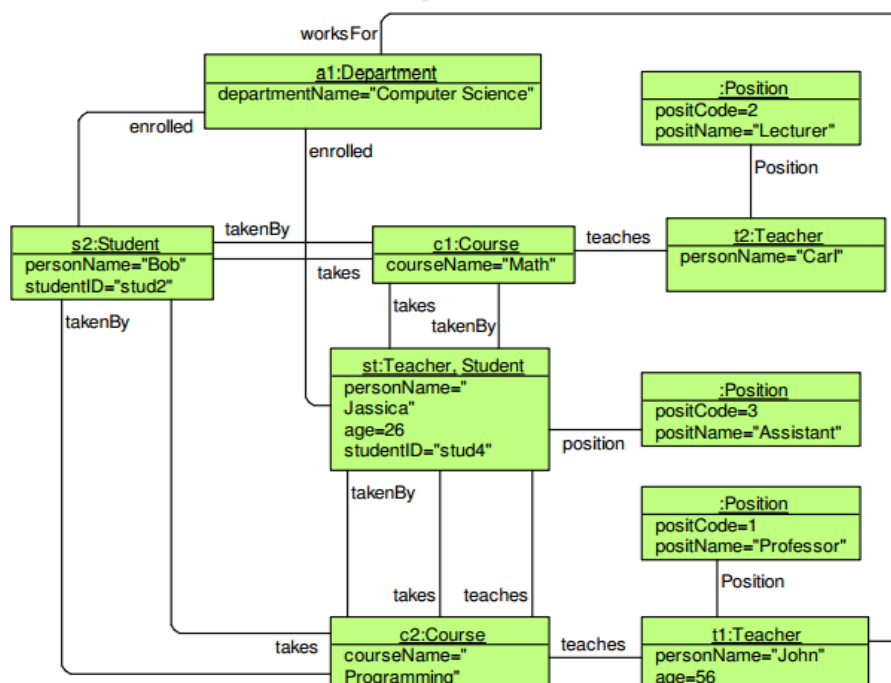
- **SPARQL** - dotazovací jazyk SPARQL¹⁰ je primárně určený k manipulaci s RDF databázemi a k tvoření sofistikovaných dotazů nad RDF grafy. SPARQL nám tedy umožňuje získávat informace z ontologií [20]. Jazyk SPARQL je pro jazyk GQL cílovou technologií.

Pro jazyk GQL byla navržena UML/OWL podmnožina. Základní myšlenka při jejím návrhu byla, že se v ní mohou používat pouze OWL DL konstrukce, které mohou být adekvátně reprezentované třídními UML diagramy. Nad výslednou podmnožinou se poté vytvořil grafický dotazovací jazyk GQL.

V kontextu s navrhovanou podmnožinou byly definovány dvě části ontologického popisu:

- **Schématická část ontologie** - obsahuje třídy, vlastnosti a relace (např.: `rdfs:subClassOf`, `rdfs:subPropertyOf`, `owl:equivalentClass`), viditelné na obrázku 7.1.
- **Datová část ontologie** - je použita k vyjádření jednoduchých tvrzení o zdrojích: datové hodnoty, pojmenované vlastnosti a třídy, které jsou

¹⁰SPARQL - Simple Protocol and RDF Query Language.



Obrázek 7.2: Datová část ontologie [3].

definované ve schématické části. Z pohledu UML obsahuje tato část instance tříd a asociací ze schématické části. Příklad datové části můžeme vidět na obrázku 7.2.

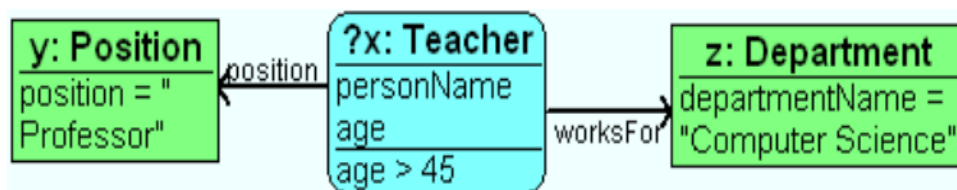
Velkou výhodou jazyka GQL je jeho kompatibilita s grafickou UML vizualizací zmiňované podmnožiny ontologií, tudíž uživatelům poskytuje kompletní grafické uživatelské rozhraní¹¹ pro průzkum schématické části ontologie a pro dotazování se na datovou část ontologie.

7.1.2 Průběh dotazování

Pro popis dotazování jazyka GQL použijeme ontologické schéma university a její odpovídající datové schéma z obrázků 7.1 a 7.2. Dotazování může probíhat různými způsoby:

1. Vybereme třídu („centrální koncept“), na kterou se budeme dotazovat. Pro názorný příklad z prostředí university můžeme použít dotaz: „*Najdi všechny učitele pozice profesor, kteří jsou starší čtyřiceti pěti let a pracují pro oddělení: Computer Science.*“ Stejný dotaz převedený do grafického

¹¹GUI - Graphical User Interface.



Obrázek 7.3: Příklad dotazu jazyka GQL [3].

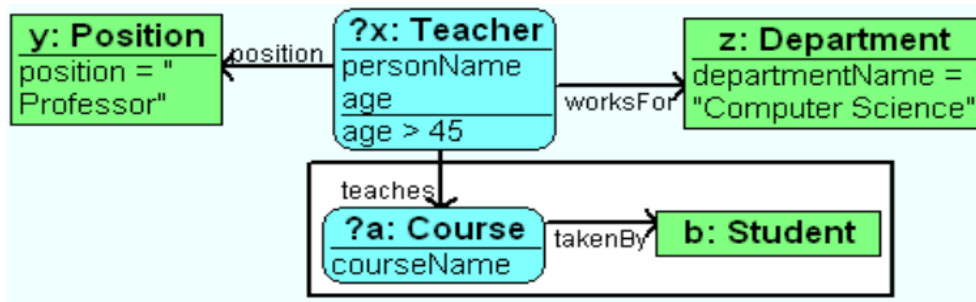
```

PREFIX uni:<http://www.owlontologies.com/University.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?personName ?age WHERE
  {?x rdf:type uni:Teacher.
  ?x uni:position ?y.
  ?y rdf:type uni:Position.
  ?x uni:worksFor ?z.
  ?z rdf:type uni:Department.
  ?x uni:age ?age.
  ?y uni:position ?position.
  ?z uni:departmentName ?departmentName.
  OPTIONAL {?x uni:personName ?personName. }
  FILTER ( ?age > 45 && ?position = "Professor" && ?departmentName =
"Computer Science")}
```

Obrázek 7.4: Přeložený dotaz z obrázku 7.3 do jazyka SPARQL [3].

formátu GQL můžeme vidět na obrázku 7.3. Tento dotaz je následně přeložen do dotazu jazyka SPARQL z obrázku 7.4.

2. Pro více komplexní dotazy můžeme použít „kontextový rám“, který je vyznačený tučnou čarou na obrázku 7.5. Tento obrázek vyjadřuje stejný dotaz jako z předchozího bodu, navíc však přidává do výstupu kurzy, které učí učitel a které jsou navštěvovány nějakým studentem. Tento rám obsahuje centrální koncept kurzu, který je orientovaný pod koncept učitele.
3. Výsledek dotazu z 2. bodu nám vrátil také jména učitelů, kteří neučí žádný kurz. Důvodem je kontextový rám vyznačený tučnou čarou, který nám říká, že obsah může být volitelný. Pokud bychom chtěli do dotazu zahrnout pouze učitele, kteří učí nějaký kurz, museli bychom použít kontextový rám vyznačený dvouvrstvou čarou, připomínající krychli. Tento rám nám zaručí, že obsah bude povinný.



Obrázek 7.5: Příklad dotazu jazyka GQL s „kontextovým rámem“ [3].

Možností pro dotazování je u jazyka GQL více, popisovat je zde dále nebudeme. Důležitá je přítomnost různých grafických prvků (centrální koncept, rámy a další) v jazyce. S uvedenými prvky můžeme pracovat a vytvářet tak dotazy.

7.2 Porovnání jazyků

V této sekci se dostáváme k porovnání nově navrženého jazyka MBIQL s jazykem GQL, který jsme si představili. Sekce je systematicky rozdělena na dvě části. V první části jsou popsány aspekty, které jsou pro oba jazyky společné, popřípadě aspekty, ve kterých můžeme vidět podobnost jazyků. Druhá část naopak popisuje hlavní rozdíly.

7.2.1 Shodné aspekty

- Oba porovnávané jazyky pracují s grafickými prvky.
- Jazyk GQL může být zvažován jako pre-processor, jelikož při tvorbě grafického dotazu na svém pozadí vytváří dotaz jazyka SPARQL. Podobné tvrzení můžeme říct i o jazyku MBIQL, pro který platí, že po dokončení tvorby jeho dotazu je tento dotaz automaticky transformován/překládán do dotazu jazyka Cypher.
- Specifikace dotazů funguje na stejném principu. Atributy jsou specifikovány/omezovány v grafických prvcích.
- Pro oba jazyky byl zvolen analogický postup návrhu. V případě jazyka MBIQL se jednalo o vytvoření nového schématu a následný návrh jazyka nad tímto schématem. U jazyka GQL došlo k vytvoření podmnožiny OWL DL, nad kterou byl navržen jazyk GQL.
- U obou jazyků se naskytuje možnost jejich budoucího rozšíření.

- Nový jazyk MBIQL byl stejně jako jazyk GQL navržen ze stejného důvodu - poskytnout uživatelům možnost intuitivnější tvorby dotazů pomocí grafických prvků.

7.2.2 Rozdílné aspekty

- Průběh dotazování je rozdílný. Zatímco u jazyka MBIQL máme přesně určené dvě metody dotazování (Specificky orientované dotazování a Dotazování pomocí vazeb), u jazyka GQL můžeme vytvářet dotaz několika způsoby, v prvním kroku pouze vybereme konkrétní třídu ze schématické části ontologie.
- Grafické prvky se u jazyků liší. Rozdílné je i jejich použití. U nového jazyka je určené použití jednotlivých prvků pro každý krok obou definovaných metod - uživatel může vybírat ve vybraných krocích pouze druhy grafických prvků (Objekt a Vazba, např.: Objekt - Úloha), ale strukturou metod je předem dané, jaké grafické prvky jsou v konkrétních krocích použity. U grafického dotazovacího jazyka GQL můžeme používat grafické prvky jazyka dle našeho uvážení.
- Cílovou technologií pro jazyk GQL je RDF databáze, pro jazyk MBIQL grafová databáze Neo4j. Tyto databáze pracují s jinými typy dat [20].
- Nový jazyk MBIQL je navržený nad schématem datového úložiště. Dotazy vzniklé v tomto jazyce jsou transformovány do Cypher dotazů, které jsou použitelné pouze pro zmiňované schéma, popřípadě schéma jemu podobné. U jazyka GQL toto omezení neplatí.

Návrh implementace

Poslední kapitola této práce bude věnována návrhu implementace, která se bude týkat dotazovacího jazyka MBIQL a s ním spojené aplikace. Nejprve si ukážeme a popíšeme modely aplikace - model nasazení a use-case model. V druhé části se dozvíme informace o návrhu koncepce dotazovacího nástroje a implementačního prostředí. Na konci kapitoly věnujeme pozornost implementační části práce. Tato část se zaměří na tzv. „proof-of-concept“ dotazovacího jazyka podložený materiály z šesté kapitoly (viz 6).

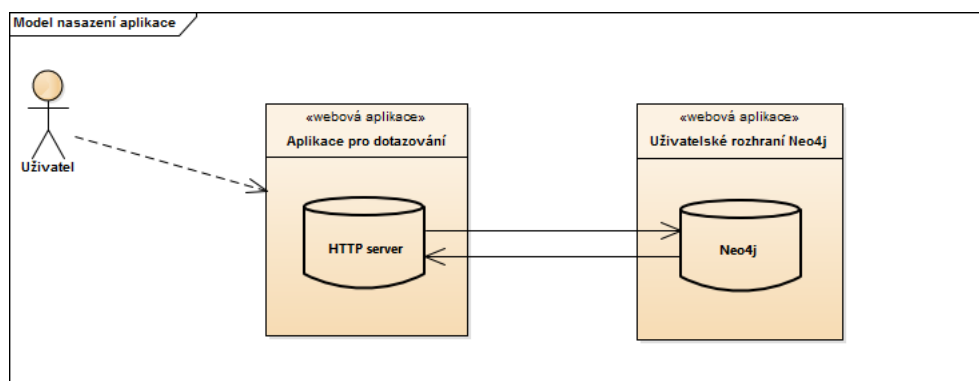
8.1 Modely aplikace

Pro aplikaci dotazovacího jazyka byly navrženy dva modely. První z nich (model nasazení) ukazuje, jak by měla výsledná aplikace fungovat po jejím nasazení. Druhý (use-case model) popisuje případy, které bude uživatel v aplikaci využívat.

8.1.1 Model nasazení

Aby bylo umožněno uživateli dotazování pomocí nového jazyka MBIQL, je nutné vyvinout aplikaci. Po její implementaci by měla být nasazena na server, na kterém poběží. Tento server by měl komunikovat s grafovou databází Neo4j, ze které bude získávat data pomocí Cypher dotazů. Komunikace by tedy měla probíhat oběma směry, kdy se server na základě zadaných požadavků od uživatele zeptá na konkrétní data a následně je od Neo4j získá zpátky.

Na obrázku 8.1 můžeme tento model vidět. Lze na něm vidět mimo jiné i webovou aplikaci: „Uživatelské prostředí Neo4j“, která je v modelu uvedena pro případ, kdy by se do ní uživatel „přepnul“ a výsledný Cypher dotaz si nechal zobrazit v tomto prostředí. Mohl by tak například využívat vlastnost Neo4j pro „procházení okolí“, o které jsme si mohli přečíst v textu 3.1.3.1 a u Specificky orientovaného dotazování, u kterého může být také využita.



Obrázek 8.1: Model nasazení aplikace pro dotazování.

8.1.2 Use-case model

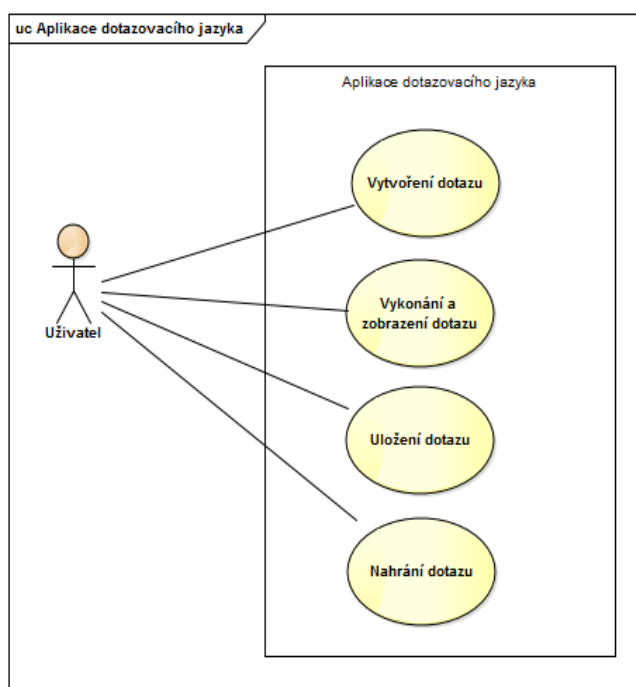
Uživatel se bude do aplikace přihlašovat za účelem získávání dat z MBI. K tomu je potřeba nejprve vytvořit dotaz, který bude zaměřen, podle struktury a specifikace dotazu, na konkrétní množinu dat uloženou ve schématu.

Uživatel bude mít různé možnosti práce s dotazy:

- **Vytvoření dotazu** - pro dotazy jazyka MBIQL. Vytvoření dotazu pomocí jedné z metod: Specificky orientované dotazování a Dotazování pomocí vazeb. Po vytvoření dotazu proběhne jeho zautomatizovaná transformace na dotaz jazyka Cypher.
- **Vykonání a zobrazení dotazu** - pro dotazy jazyka MBIQL a Cypher dotazy. Vykonání se bude týkat jen Cypher dotazů, ve smyslu provedení dotazu nad databází Neo4j a získávání dat. Zobrazování dotazů bude umožněno jak pro dotazy nového jazyka, kdy uživatel uvidí svojí „cestu“ při vytváření dotazu, tak pro výsledné Cypher dotazy.
- **Uložení dotazu** - ukládání Cypher dotazů umožní Knihovna pro práci s Cypher dotazy, viz obrázek 8.3.
- **Nahrání dotazu** - nahrání Cypher dotazů umožní Knihovna pro práci s Cypher dotazy, viz obrázek 8.3.

Pro Uložení a Nahrání dotazů nového jazyka („cesty“ těchto dotazů) by mohla být použita speciální knihovna pro práci s těmito dotazy.

Use-case model je graficky znázorněn na obrázku 8.2.



Obrázek 8.2: Use-case model dotazovací aplikace.

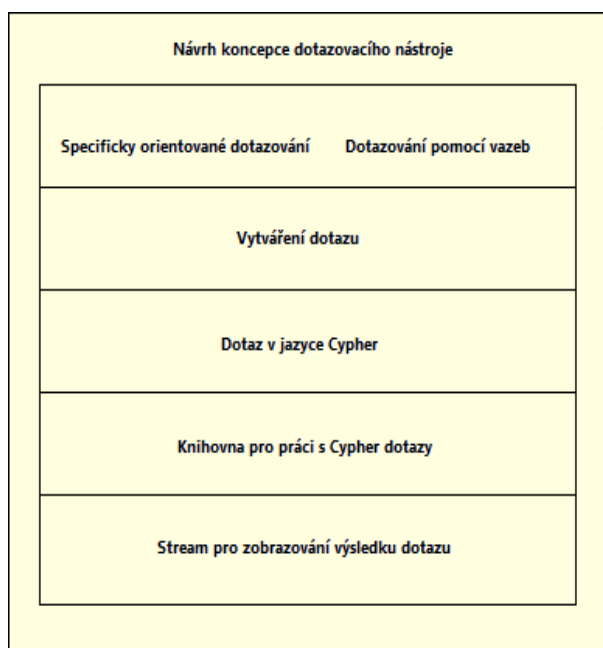
8.2 Návrh koncepce a implementační prostředí

Před začátkem implementace je důležité vytvořit návrh koncepce dotazovacího nástroje, čili nové aplikace pro dotazování. Po tomto návrhu nastíníme, jaké implementační prostředí bychom měli použít při samotném vývoji.

8.2.1 Návrh koncepce dotazovacího nástroje

Při pohledu na obrázek 8.3, který ukazuje návrh koncepce dotazovacího nástroje, zjistíte, že by výsledná aplikace měla být rozdělena na pět hlavních částí:

- **Vybrání dotazovací metody** - horní část aplikace bude sloužit k rozhodnutí o dotazovací metodě jazyka MBIQL. Uživatel má na výběr ze Specificky orientovaného dotazování a Dotazování pomocí vazeb.
- **Vytváření dotazu** - místo určené pro vytváření dotazu, které je uzpůsobeno zvolené dotazovací metodě. Pro Specificky orientované dotazování budou viditelné pouze Objekty bez jakýchkoliv Vazeb. Dotazování pomocí vazeb bude zobrazovat Objekty i Vazby ve schématu podobnému modelu MBI z obrázku 1.1. Ve schématu oproti modelu MBI přibudou nové Vazby mezi hierarchiemi objektů a ke každé Vazbě bude přidán její



Obrázek 8.3: Návrh koncepce dotazovacího nástroje (aplikace).

směr. Ostatní grafické prvky obou metod budou zobrazovány na základě průběhu metod. Finální vzhled těchto prvků bude navržen autorem celkové implementace dotazovacího jazyka MBIQL.

Při vytváření dotazu by se měly pro uživatele objevovat upozornění, která po vybrání většího množství prvků (Objekt a Vazba) v cyklu nebo atributů na vybraném prvku oznámí, že by měl uživatel pokračovat na další krok dotazování. Další možností je automaticky omezit dotazování po určitém počtu zvolených prvků či atributů - pokračovat na další krok v metodě.

- **Dotaz v jazyce Cypher** - místo vybrané pro zobrazování Cypher dotazu, který byl vytvořen transformací dotazu jazyka MBIQL. Může se také jednat o Cypher dotazy, které byly uživatelem nahrány do aplikace.
- **Knihovna pro práci s Cypher dotazy** - umožňuje ukládání a nahrávání Cypher dotazů.
- **Stream pro zobrazování výsledku dotazu** - zobrazování výsledného grafu získaného z datového úložiště. Pro uživatele by měla být poskytnuta možnost „přepnutí“ do Uživatelského prostředí Neo4j.

8.2.2 Implementační prostředí

Implementační prostředí vybereme podle požadavků na dotazovací aplikaci, z nichž některé vyplynuly z představovaných modelů a ostatních informací uváděných v této práci:

- Ovládání interaktivních grafických prvků.
- Možnost ukládání a nahrávání dotazů Cypher pro uživatele.
- Podpora protokolu HTTP.
- Přihlašování uživatelů.
- Transformace dotazů.
- Modul pro práci s Neo4j - posílání Cypher dotazů, ukládání výsledku.

Na frontend (část, kterou vidí uživatel) by měly být použity jazyky:

- **HTML** - zajistí základní strukturu stránek v aplikaci.
- **CSS** - zajistí vizuální stránku aplikace.
- **JavaScript** - zajistí „dynamicitu“ stránek aplikace (interaktivní ovládání dotazování).

Pro backend můžeme použít více jazyků. Důležité však je, aby splňovaly požadavky na dotazovací aplikaci, kterou má zařídit serverová část (komunikace s Neo4j, podpora HTTP, přihlašování atd.).

Navrhované možnosti řešení: PHP Framework, Java a další.

8.3 Proof-of-concept

Jako důkaz realizovatelnosti dotazovacího jazyka MBIQL „odsimulujeme“ jeden ze vzorových dotazů z textu 6.1.2. V průběhu šesté kapitoly (viz 6) jsme prošli většinu vzorových dotazů vytvořených pro návrh jazyka. Vybereme proto jeden z dotazů, u kterého jsme průběh metodou nepopisovali dopodrobna. Dotaz se pokusíme rozebrat do větších detailů.

8.3.1 Simulace vzorového dotazu

4. *Zjistí, do jaké Skupiny metrik patří Metrika: „Počty spravovaných technických prostředků“.* Zobraz všechny hrany a uzly.

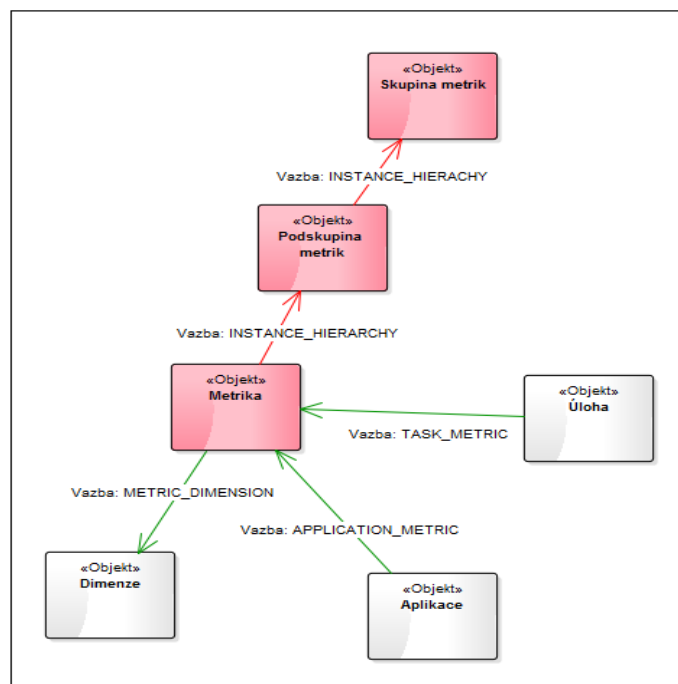
- **Výběr Objektu** - začíná Struktura dotazu. V prvním kroku zvolíme počáteční Objekt: Metrika z celého schématu modelu. Část schématu důležitá pro tento dotaz je vidět na obrázku 8.4. Zvolili jsme tedy první prvek. Uložíme si do proměnné hodnotu, která bude mít pro Cypher dotaz význam: (a:METRIC). V dalších krocích bude docházet k automatickému ukládání vybraných prvků a hodnot do proměnných, nebudeme proto o tomto procesu dále psát.
- **Výběr Vazby (1)** - vybereme první Vazbu: INSTANCE_HIERARCHY, která vede z počátečního Objektu: Metrika na Objekt: Podskupina metrik. Uživateli bylo umožněno vybrat, kromě této Vazby, Vazby označené zelenou barvou na obrázku 8.4. Mezi nově uložené grafické prvky patří Vazba: (a)-[relMetricsoupgroup:INSTANCE_HIERARCHY]->(b) a Objekt: (b:METRICSUBGROUP).
- **Výběr Vazby (2)** - uživatel si přeje pokračovat. Vybírá si proto další Vazbu, tentokrát Vazbu: INSTANCE_HIERARCHY mezi Podskupinou metrik a Skupinou metrik. Dojde ke zvýšení počtu prvků. Tato přidaná Vazba: (b)-[relMetricgroup]->(c) a Objekt: (c:METRICGROUP). Uživatel se následně rozhodne, že už nechce vybírat další prvky. Dotazováním vybral prozatím pět prvků: tři Objekty a dvě Vazby (označeny růžovou a červenou barvou na obrázku 8.4).
- **Rozhodnutí o zobrazení prvků modelu (Uzlů typu M)** - uživatel se v tomto případě rozhodne, že nežádá o zobrazení prvků modelu. Nebudou přidány žádné nové prvky. Struktura dotazu je u konce.
- **Výběr prvků ze Struktury dotazu** - uživatel zvolí v Příznakovém poli, ve kterém bude pět položek (podle počtu zvolených prvků ze Struktury dotazu) pouze položku přiřazenou k Objektu: Metrika.
- **Zvolení atributu na prvku** - uživatel zvolí pouze jeden atribut ze Seznamu, kde se mu zobrazí atributy entity Uzel, která má atribut OBJECT_TYPE = METRIC, kromě tohoto atributu. Uživatel vybere atribut Název (NAME).

Význam pro Cypher dotaz: WHERE a.NAME

- **Zadání hodnoty atributu** - uživatel má dvě možnosti: zadání hodnoty do Textového pole, nebo výběr hodnoty ze Seznamu (seznam všech názvů instancí objektu Metrika z datového úložiště). Uživatel zvolí například druhou možnost a vybere ze Seznamu hodnotu: „Počty spravovaných technických prostředků“.

Význam pro Cypher dotaz: WHERE a.NAME = 'Počty spravovaných technických prostředků'

Uživatel je v tomto případě spokojený s omezením jednoho atributu.



Obrázek 8.4: Schéma ke vzorovému dotazu číslo 4.

- **Výběr návratových hodnot** - výběr z Příznakového pole pěti prvků. Uživatel zvolí všechny prvky.

Význam pro Cypher dotaz: RETURN a, relMetricsubgroup, b, relMetricsubgroup, c;

Po transformaci dotazu nového jazyka vznikne finální Cypher dotaz:

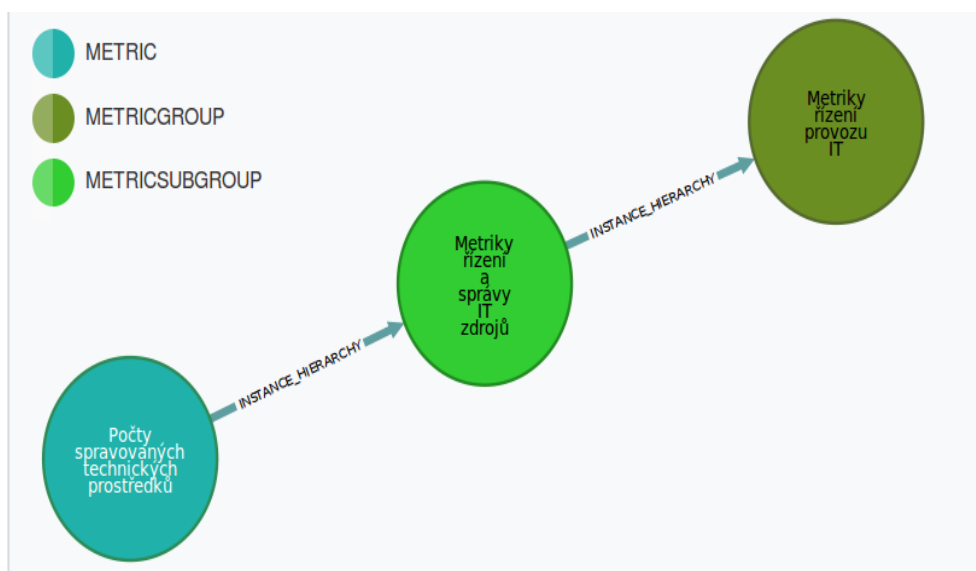
```

MATCH (a:METRIC)-[relMetricsubgroup:INSTANCE_HIERARCHY]->
(b:METRICSUBGROUP),
(b:METRICSUBGROUP)-[relMetricgroup:INSTANCE_HIERARCHY]->
(c:METRICGROUP)
WHERE a.NAME = 'Počty spravovaných technických prostředků'
RETURN a, relMetricsubgroup, b, relMetricgroup, c;

```

Aliases pro grafické prvky (Objekty a Vazby) jsou pouze ukázkové. Všimněte si, že finální Cypher dotaz se liší od výsledného Cypher dotazu uváděného v textu 6.3.2.2 k tomuto dotazu. V tomto řešení je uváděn u Vazeb atribut LABEL. Výsledek dotazu však vrací totožnou množinu dat. Automatizovaný proces transformace by měl pro jistotu doplňovat ke zvoleným Objektům jejich OBJECT_TYPE, k Vazbám jejich LABEL, a ne jen jednu variantu.

Po vykonání Cypher dotazu na straně Uživatelského prostředí Neo4j se uživateli zobrazí uzly a hrany viditelné na snímku 8.5.



Obrázek 8.5: Výsledek dotazu číslo 4.

Závěr

Cílem práce bylo zanalyzovat uživatelské potřeby pro dotazování se na vztahy MBI objektů, návrh dotazovacího jazyka nad těmito objekty a návrh implementační části jazyka. Práce byla systematicky rozdělena do osmi kapitol, které měly pokrýt všechny body zadání práce a splnit výše zmiňovaný cíl.

Nejprve došlo k důležitému seznámení s informační bází MBI, které nás zasvětilo do problematiky. V první kapitole (viz 1) byla také představena bakalářská práce: „Vizualizace vztahů v informační bázi MBI“, jejímž autorem je Bc. Vojtěch Stránský a na kterou bylo v této práci navázáno.

V druhé části (viz kapitola 2) jsme provedli analýzu uživatelských potřeb/požadavků. V analýze byly rozebrány skupiny uživatelů, uživatelských dotazů a proběhla analýza současného stavu portálu. Z analýzy portálu vyplynuly jeho výhody a nedostatky, které byly v kapitole také popsány. Na jejím konci byla analýza vyhodnocena.

V textu o cílových technologiích (viz kapitola 3) byly popsány grafové databáze s hlavním představitelem - Neo4j. Grafová databáze Neo4j byla zařazena mezi cílové technologie, protože je datovým úložištěm dat z MBI. Do těchto technologií byl vybrán také jazyk Cypher, který je dotazovacím jazykem databáze Neo4j a navrhovaný jazyk je do něj překládán.

Ve čtvrté a páté kapitole (viz kapitoly 4 a 5) byly odděleně popsány dvě schémata datového úložiště - staré a nové schéma. Staré schéma vzniklo ještě před tvorbou této práce a bylo potřeba se s ním seznámit, jelikož na jeho základech bylo poté navrženo nové schéma. Toto schéma nám dalo vhodný podklad pro nový jazyk.

Návrh nového jazyka, pojmenovaného MBIQL, pak proběhl v šesté kapitole (viz 6). Cílem této kapitoly bylo navrhnout doménově specifický dotazovací jazyk, který bude graficky orientovaný, tak totiž umožňuje uživateli intuitivnější ovládání. Dotazovací jazyk byl rozdělen na dvě metody - Specificky orientované dotazování a Dotazování pomocí vazeb, pro které byly vytvořeny vzorové dotazy. Toto rozdělení vyplynulo z prováděných analýz. Vzorové dotazy byly v průběhu definování metod „odsimulovány“. Na konci kapitoly bylo

nastíněno Asistované procházení grafem, které s jazykem souvisí.

V předposlední kapitole (viz 7) byl po průzkumu autora práce zvolen grafický dotazovací jazyk, který byl ve stručnosti představen a následně porovnán s novým jazykem MBIQL. Pro poslední kapitolu (viz 8) bylo vybráno téma návrhu implementace. V této části byl popsán návrh budoucí aplikace pro dotazování pomocí nového jazyka. Pro tuto aplikaci bylo zvoleno vhodné implementační prostředí. Na samotném konci kapitoly byla prokázána realizovatelnost jazyka MBIQL formou „proof-of-concept“ jednoho ze vzorových dotazů.

Cíl práce byl splněn. Dotazovací jazyk se povedlo úspěšně navrhnout a prokázalo se, že tento návrh bude realizovatelný. Pro celou komunitu MBI je vytvořen dobrý základ pro alternativní způsob dotazování. V budoucnu je však nutné navrhovanou aplikaci kompletně implementovat a následně otestovat. Toto by mohl být vhodný návrh pro bakalářskou/diplomovou práci.

Po otestování mohou vyplynout některé nedostatky nového jazyka, které by mohly být v budoucnu opraveny a jazyk by tak mohl být rozšířen. O rozšíření jazyka můžeme uvažovat i v případě, kdy žádné nedostatky nebudou, ale přijde žádost ze strany MBI, popřípadě od jejich uživatelů.

Osobním přínosem pro autora práce je seznámení s cílovými technologiemi, které jsou velkým příslibem do budoucna. Návrh nového jazyka a vytváření rozsáhlé práce daly autorovi zcela nové zkušenosti. Opomenuty nemohou zůstat ani získané informace o MBI a podnikové informatice.

Literatura

- [1] Neo Technology, Inc.: *What is a Graph Database? [online]*. [cit. 2015-03-19]. Dostupné z: <http://neo4j.com/developer/graph-database/>
- [2] Stránský, V.: *Vizualizace vztahů v informační bázi MBI*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2014. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/stranvo1_2014bach.pdf
- [3] Barzdins, G.; Rikacovs, S.; Zviedris, M.: *Graphical Query Language as SPARQL Frontend [online]*. [cit. 2015-05-03]. Dostupné z: http://www.semti-kamol.lv/doc_upl/ADBIS_final.pdf
- [4] Voříšek, J.; Pour, J.; kolektiv: *Management podnikové informatiky*. Praha: Professional Publishing, první vydání, 2012, ISBN 978-80-7431-102-34.
- [5] Pour, J.: *MBI – Management Byznys Informatiky [online]*. [cit. 2015-03-01]. Dostupné z: <http://mbi.vse.cz>
- [6] MBI tým: *MBI, Management Byznys Informatiky – Koncepce a návod k použití [online]*. [cit. 2015-03-05]. Dostupné z: <http://mbi.vse.cz/mbi/files/help.pptx>
- [7] Rouse, M.: *Definition, graph database [online]*. 2014, [cit. 2015-03-16]. Dostupné z: <http://whatis.techtarget.com/definition/graph-database>
- [8] Holý, J.: *Grafové databáze a Neo4j [video]*. 2012, [cit. 2015-03-16]. Dostupné z: <https://www.youtube.com/watch?v=lLuqIcYvWUQ>
- [9] Sherman, M.: *Your Database Is So Retro: Old Data, New Databases [online]*. 2014, [cit. 2015-03-16]. Dostupné z: <http://readwrite.com/2011/04/20/5-graph-databases-to-consider>

- [10] Kotowski, J.: *Sémantické technologie: Grafové databáze [online]*. 2013, [cit. 2015-03-21]. Dostupné z: <http://computerworld.cz/technologie/semanticke-technologie-grafove-databaze-50582>
- [11] Ramba, J.: *Přehled grafových databází [online]*. 2013, [cit. 2015-03-16]. Dostupné z: <http://www.neo4j.cz/>
- [12] Finley, K.: *5 Graph Databases to Consider [online]*. 2011, [cit. 2015-03-16]. Dostupné z: <http://readwrite.com/2011/04/20/5-graph-databases-to-consider>
- [13] Bachman, M.: *Informatický večer FIT - Grafové databáze (první oficiální Neo4j Meetup) [video]*. 2014, [cit. 2015-03-19]. Dostupné z: <https://www.youtube.com/watch?v=YL1IyWzDvK0>
- [14] Neo Technology, Inc.: *Intro to Cypher [online]*. [cit. 2015-03-21]. Dostupné z: <http://neo4j.com/developer/cypher-query-language/>
- [15] Taylor, A.; Jones, A.: *Cypher Query Language [online]*. 2012, [cit. 2015-03-21]. Dostupné z: <http://www.slideshare.net/graphdevroom/cypher-query-language>
- [16] Neo Technology, Inc.: *The Neo4j Manual - What is Cypher? [online]*. [cit. 2015-03-21]. Dostupné z: <http://neo4j.com/docs/stable/cypher-introduction.html>
- [17] Hunger, M.: *Query with Cypher [video]*. 2012, [cit. 2015-03-21]. Dostupné z: http://www.neo4j.org/tracks/cypher_track_start
- [18] Štencek, J.: *Kapitola 3. Principy sémantického webu: Ontologie [online]*. [cit. 2015-05-03]. Dostupné z: <http://vse.stencek.com/semanticky-web/ch03s05.html>
- [19] Zendulka, J.: *Projektování programových systémů - 7 Jazyk UML (Unified Modeling Language) [online]*. [cit. 2015-05-03]. Dostupné z: http://www.fit.vutbr.cz/study/courses/PPS/public/pdf/7_umlch.pdf
- [20] Štencek, J.: *Kapitola 3. Principy sémantického webu: RDF [online]*. [cit. 2015-05-03]. Dostupné z: <http://vse.stencek.com/semanticky-web/ch03s04.html>

Seznam použitých zkratk

MBI Management of Business Informatics

IT Information Technology

RACI Responsible, Accountable, Consulted, Informed

GB Gigabyte

ICT Information and Communication Technologies

XML Extensible Markup Language

CSV Comma-Separated Values

ITIL Information Technology Infrastructure Library

PDF Portable Document Format

NoSQL Not Only Structured Query Language

MySQL My Structured Query Language

API Application Programming Interface

SPARQL Simple Protocol and RDF Query Language

SQL Structured Query Language

SPARQL Simple Protocol and RDF Query Language

IO Input/Output

KGI_KPI Key Goal Indicators, Key Performance Indicators

II Instance-Instance

IH Instance-Hierarchy

A. SEZNAM POUŽITÝCH ZKRATEK

IM Instance-Model

HTML HyperText Markup Language

MBIQL Management of Business Informatics Query Language

GQL Graphical Query Language

OWL Web Ontology Language

OWL DL Web Ontology Language Description Logic

UML Unified Modeling Language

RDF Resource Description Framework

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

CSS Cascading Style Sheets

PHP PHP: Hypertext Preprocessor

Uživatelská příručka a další přílohy

B.1 Uživatelská příručka

Uživatelská příručka se týká konfigurace uživatelského rozhraní grafové databáze Neo4j a je upravenou verzí. Původní verzi uvedl ve své bakalářské práci Bc. Vojtěch Stránský.

Nová verze obsahuje nové a aktualizované údaje z původní verze a zaměřuje se na připojení ke vzdálenému serveru, na kterém běží databáze Neo4j. Stará verze rozebírá spuštění lokálního serveru.

B.2 Konfigurační soubory

- **Konfigurace uzlů** - slouží ke konfiguraci uzlů v uživatelském prostředí Neo4j.
- **Konfigurace vlastností** - slouží ke konfiguraci vlastností v uživatelském prostředí Neo4j.

Oba tyto soubory jsou vytvořeny tak, aby seděly na nové schéma datového úložiště.

B.3 Dotazy

- **Upravené dotazy** - upravená verze dotazů (jejich Cypher verze), které byly vytvořeny v bakalářské práci „Vizualizace vztahů v informační bázi MBI“ - autor Bc. Vojtěch Stránský. Cypher dotazy jsou upraveny tak, aby vracely smysluplné výsledky pro nové schéma.

B. UŽIVATELSKÁ PŘÍRUČKA A DALŠÍ PŘÍLOHY

- **Vzorové dotazy** - dotazy (jejich Cypher verze), které byly vytvořeny v této práci při návrhu nového jazyka MBIQL.

Všechny přílohy jsou obsaženy na přikládaném CD, viz dále.

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
manual	
├─ Upravené_dotazy.cql	upravené dotazy Bc. Vojtěcha Stránského
├─ Uživatelská_příručka.docx...	uživatelská příručka ve formátu docx
├─ Uživatelská_příručka.pdf	uživatelská příručka ve formátu pdf
├─ Vzorové_dotazy.cql	vzorové dotazy
src	
├─ Konfigurace_uzlů.cql	konfigurace uzlů v prostředí Neo4j
├─ Konfigurace_vlastností.properties	konfigurace vlastností v prostředí Neo4j
├─ thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
├─ BP_Batík_Ondřej.pdf	text práce ve formátu PDF