**ZT1**

**Data –** údaje popisující realitu, forma je textová, kódová či obrazová. Data se získávají z DB pomocí SQL dotazů, cílem je data přenést, zpracovat či zpřístupnit.

**Informace** – Data, která pro nás mají význam a smysl. Cílem je snížit entropii (nejistotu). Je to surovina pro tvorbu znalostí.

**Znalosti –** Informace nebo data začleněná do souvislostí, cílem je porozumění těmto souvislostem

**Znalostní technologie -** SW prostředky, které operují se znalostmi, jak je získávat, sdílet, uchovávat, zpracovávat, předávat a vytvářet.

**Syntaktický web –** v systému je problematické klást komplexní dotazy využívající znalosti, získávat znalosti z různých datových zdrojů a delegovat komplexní úkoly webovým agentům. Tyto problémy se řeší přes XML, jakožto značkovací jazyk.

**Problémy syntaktického webu:**

Nelze klást komplexní dotazy (Najdi zverimex v HK, který má otevřeno….)

Získávat znalosti z různých datových zdrojů (Hledej takové dopravní spojení, abych se dostal z NBK do kina v HK za půl hodiny před filmem Avatar).

**XML –** eXtensible Markup Language, značkovací jazyk, jehož účelem je označkování textových dat pomocí vlastních tagů a dodání jasné struktury webovým dokumentům. Je dobře zpracovatelný lidmi i stroji.

**Ontologie z pohledu počítačové vědy –** Ontologie je explicitní (jednoduše dostupná) specifikace konceptualizace (systém pojmů modelující určitou část světa).

**Vrstvy** – XML, RDF, Ontologie

**TBox –** slovník (taxonomie) obsahující definice konceptů a vztahů mezi nimi

**ABox –** konkrétní fakta (data) vztažená k určitému slovníku (TBoxu)

**Použití OWL –** OWL jako jazyk pro výměnu dat, pro tvorbu ontologií, založený na deskripční logice, OWL jako terminologie nebo znalostní model

**Přínos Ontologie –** Komunikace člověk – člověk: jednoznačná, komunikace znalostní inženýr- expert, podpora sběru požadavků pro IS

Komunikace PC – PC: podpora komunikace mezi různými platformami podle jednotného slovníku

Komunikace člověk – PC: znovupoužitelnost (formální zakódování důležitých entit a atributů, ontologie použitelná v různých aplikacích s možností přizpůsobení), Vyhledávání, Spolehlivost

Ontologie zachycuje význam nějaké specifické oblasti nebo znalosti, odpovídá tomu, co člověk ví a zná o určité oblasti.

**Normalizovaná ontologie** – ontologie, kterou systematickým způsobem strukturujeme tak, že na nejvyšší úrovni jsou tři třídy – primitivní, přídavné a definované

**Složky jazyka pro reprezentaci ontologií**

**Syntaktická** –každý jazyk má syntaxi, syntaktické výrazy jsou bez významu, dokud jim nepřiřadíme objekty modelu.

**Strukturální –** způsob organizace prvků modelu, např. databázové schéma, tabulky, řádky, sloupce…

**Sémantická – p**řiřazení mezi strukturovanou podmnožinou dat a modelem množiny objektů. Tento model existuje jen v naší mysli. Znát sémantickou interpretaci = umět cizí jazyk

**Pragmatická –** co vše lze z dané informace vyčíst. Bude pršet -> budu mokrý

**Cíle sémantického webu -** Dodat informacím na webu jednoznačně reprezentovaný význam, zachycený v jazyku přístupném pro automatické zpracování. Automatické nástroje pracující se znalostmi budou schopné do velké míry samostatně nalézat relevantní zdroje informací, zpracovávat je a odvodit ze stávajících znalostí nové.

**Ontologie a její součásti**

**Inverzní vlastnost** – přiřazení, které vede i opačným směrem, např. inverzní vlastností k HK hasPart Malšovice je Malšovice isLocatedIn HK

**Tranzitivní vlastnost** – vztah X je tranzitivní, jestliže platí:

Když A je ve vztahu X k B, a B je ve vztahu X k C, pak A je ve vztahu X k C

**Symetrická vlastnost** – pokud platí A je ve vztahu X k B a B je ve vztahu X k A, např. sourozenci

**Funkcionální vztah** – pro každé x existuje max. jedno f(x)

**Odvozování klasifikátorem –** kontrola konzistence dat, klasifikace tří a jedinců

Př.

máme třídu Samotář, která je definována takto:každá taková osoba, která má nejvýše jednoho důvěrného přítele. Víme, že Jeroným má přítele Lea, máme popsat Jeronýma tak, aby ho klasifikátor zařadil jako samotáře musíme zjistit, zda nemá nějaké další přátele, zda je Leo důvěrným přítelem nebo není, zda je Jeronym osoba, zda je Leo osoba, zda je vztah funkcionální

**Reflexivní vlastnost –** Vztah X je reflexivní, jestliže pro jedince A platí: A je ve vztahu k A, Df musí mít neprázdný průnik. Př. větší nebo rovno, dělitelný na množině přirozených čísel.

**Datotypová vlastnost –** Přiřazuje jedincům primitivní hodnoty. Př. Jan máVěk 22.

**Třída –** množina jedinců se společnými charakteristikami

**Definiční obor –** může být i několik tříd, jedinec se může účastnit pouze takového vztahu, kde splňuje Df.

př. Hradec (destinace) máUbytování Hotel Černigov (ubytování)

Df – destinace, Obor hodnot – ubytování

**Obor hodnot –** Určuje, jakých hodnot může jedinec u dané vlastnosti nabývat, může být i několik tříd

**Nadtřída a podtřída –** třídy jsou uspořádány do hierarchie podle principu generalizace-specializace. Jednotlivé třídy se mohou mít společné jedince. př. Střední Čechy -> Praha

**Disjunktnost tříd –** teoreticky se mohou všechny třídy překrývat, ale často chceme zaručit, že nemají společného jedince.

**Role –** role určuje zařazení jedince v hierarchii tříd.

**Sémantika v ontologii**

**Typy tříd –** Anonymní – bezejmenná třída vymezená definováním logických výrazů (OR, AND, výčtem), omezením.

Primitivní – je pojmenovaná, z názvu je jasné, co reprezentuje, třídu můžeme dále popsat a uvést její charakteristiky. Pizza má přísadu

Definovaná – třída vymezená definováním, třída lze definovat několika způsoby

Hlasitý papoušek – exotický pták, který má vysoký stupeň hlasitosti

**Vymezení třídy popsáním –** popsání = vymezení třídy použitím jejích charakteristik. Př. Každá pizza obsahuje přísadu -> Pizza hasTopping some Topping.

Př. Každý časopis Science American vychází každý měsíc a zabývá se přírodními vědami

-> Journal AND hasReleaseOn some Month AND hasInterest some NaturalScience

Popsání třídy nestačí pro jednoznačné zařazení jedince do třídy. K třídě musíme přidat další vlastnosti nebo změnit její typ. Třídu popisujeme, abychom mohli zkontrolovat, zda jedinec splňuje kritéria pro zařazení do dané třídy.

**Vymezení třídy definováním –** pro jednoznačnější zařazení jedince do třídy určujeme definici třídy.

Př. Každá sýrová pizza má sýrovou přísadu -> CheesePizza hasTopping some CheeseTopping

Definice třídy CheesePizza – jestliže je něco sýrová pizza, pak patří do třídy Pizza a zároveň patří do třídy všech jedinců, kteří obsahují sýrovou přísadu. Jestliže je něco Pizza a obsahuje sýrovou přísadu, pak patří do třídy sýrová pizza.

**Vymezení třídy výčtem –** třídu tvoří výčet jedinců. Př. Programovací jazyk – Java, C++, VB…

**Omezení someValuesFrom (karidální) –** alespoň jedna vlastnost musí být uvedeného typu. Př. Sýrová pizza je taková, která má alespoň jednu přísadu sýrovou

**Omezení allValuesFrom –** Všechny hodnoty musí být uvedeného typu. Ze splnění allValuesFrom nevyplyne someValuesFrom

**Omezení hasValue –** přiřazuje konkrétního jedince/hodnotu. Podobné omezení someValuesFrom, ale pro jedince a primitivní datové hodnoty

**Sjednocení –** jedinci, kteří patří do třídy A nebo B

**Průnik –** jedinci, kteří patří do A, a zároveň do B

**Doplněk –** třída jedinců, kteří nepatří do žádné třídy. Př. nevegetariánská pizza je pizza, která není vegetariánská

**Odvozovač (klasifikátor) –** usuzovací program, který je schopen z ontologie odvodit informace, které v ní nejsou explicitně vyjádřeny. Důvody použití: Sémantický web usiluje o více inteligence při strojovém zpracování informací bez asistence člověka. Pomáhají uživateli udržovat kvalitní ontologie (smysluplné, korektní, minimálně redundantní). Např. Pellet

Jeho úlohy: kontrola konzistence ontologie, klasifikace tříd, klasifikace jedinců

**Princip uzavřeného světa CWA –** Nějaké tvrzení může být pravdivé, nepravdivé nebo nerozhodnutelné. V uzavřeném světě nepředpokládáme zjišťování nových skutečností. Jestliže neumím u tvrzení X prokázat, že je pravdivé, je nepravdivé.

**Princip otevřeného světa OWA –** V otevřeném světě předpokládáme, že díky vědě a výzkumu přibudou další informace k aktuálnímu stavu. Jestliže tvrzení X není pravdivé, může být jak nepravdivé, tak nerozhodnutelné.

**Porovnání OWA vs. CWA –** Databáze je příkladem uzavřeného světa. Pokud osoba není v živnostenském rejstříku, podle CWA je osoba nenalezena, podle OWA osoba není živnostník.

**Axiom pokrytí třídy –** do třídy nebude možné přidávat další možné hodnoty, dle principu CWA.

**Binární relace –** vztah mezi objektem a maximálně jednou hodnotou.

**N-ární relace –** vztah mezi objektem a minimálně dvěma dalšími hodnotami

**Pseudojedinec –** třída bez podtříd, pseudojedinec je podtřídou té třídy, do které by měl nahrazovaný jedinec patřit.

**Normalizace –** technika, která nám pomáhá pomocí klasifikátoru uchovávat ontologii v udržitelném a modulárním stavu. Ontologii vytváříme jako stromovou strukturu nikoli jako graf

**Role –** určuje zařazení jedince na pozici v hierarchii. Role ovlivňuje, jaké typy podmínek budou u daného konceptu přiřazené.

**Meronymický vztah –** vztah mezi částí a celkem. Kompozice/agregace. Snižuje složitost, protože místo mnoha objektů se zabýváme jen celkem. Kompozice -> Hlava je částí těla.

Základní vlastnosti vztahu kompozice – Konfigurace, stejnorodost, neměnitelnost

6 typů meronymického vztahu – komponenta-jednotný objekt, materiál-objekt, porce-objekt, místo- oblast, člen-shluk, člen-partnerství.

Př. Automobil je z části železo, vztah materiál-objekt -> železo nejde od auta oddělit

Auto – kola -> komponenta, jednotný objekt, lze oddělit

**Ontologie jako axiomatický systém –** Axiomy: pravdivá tvrzení

Usuzovací pravidla: logicky korektní zásady vytváření pravdivých tvrzení

Věty: hypotézy, které je třeba dokázat

Logická teorie: axiomy+usuzovací pravidla + pravdivé věty

**Způsoby interference pomocí:**

**dedukce –** logické usuzování, při kterém musí závěr plynout z předpokladů

**indukce –** zobecnění speciálních případů. Hledáme zákonitosti spojující fakta

**abdukce –** známe B, A=>B a usuzujeme na A, hledáme vysvětlení pro fakta, která máme k dispozici

**heuristiky –** pravidla založená na zkušenosti experta

**analogie –** odvození závěru na základě podobnosti s jinou situací

**defaultních hodnot –** usuzování pomocí obecných znalostí, pokud nejsou speciální

**nemonotonního usuzování –** předcházející znalosti se mohou revidovat na základě nových poznatků

**generování a testování –** metoda pokusů a omylů, opakovaně se generuje možné řešení a testuje se, zda vyhovuje požadavkům.

**Jazyk SWRL – Pravidlový jazyk** Semantic Web Rule Languague. Rozšiřuje OWL o možnost tvorby pravidel. Odvozování vztahů: patří do

**SWRL a jeho komponenty –** Třída Film, FilmZanruSciFi, objektová vlastnost maZanr.

Film(?film)^maZanr(?film, scifi)

**OWL vs. SWRL –** OWL nepracuje s proměnnými, SWRL ano. V SWRL lze zkoumat, zda jedinec pracuje sám se sebou. U jedinců OWL nelze použít axiom uzávěru vlastnosti, ale ani u SWRL. SWRL zatím neposkytuje dostatečné prostředky pro tvorbu složitějších pravidel. SWRL neříká, jak odvozovat, je říká, jaká pravidla dodržovat.

**Dle předmětu konceptualizace –** generické, doménové, úlohové, aplikační a reprezentační onotologie.

**Generické ontologie –** širší záběr než doménové, zachycení obecných zákonitostí, nejdou příliš do hloubky. Využívají se pro tvorbu nových ontologií, reprezentaci běžného uvažování.

**Doménové ontologie –** zaměření na konkrétní věcnou oblast nebo její dílčí část. Vznikají buď jako nové ontologie, či jako obohacení existujícího informačního zdroje.

**Úlohové ontologie –** slovník pojmů, který se týká řešeného problému, ontologie jako model pro řešení problému. Aplikace např. jako diagnostika, monitorování

**Aplikační ontologie –** nejčastější typ ontologií, součást konkrétní aplikace. Obsahuje definice potřebné k modelování znalostí vyžadovaných konkrétní aplikací. Rozdělení na část obecnou a specifickou

**Reprezentační ontologie –** definuje jazyk pro reprezentaci znalosti. Ontologie definující reprezentaci abstraktního modelu znalosti.

**Taxonomie –** hierarchická doménová struktura obsahující hierarchické vztahy a jednotlivé modelované koncepty. Neexistuje standardizovaný přístup pro modelování ani definice. Používá se hlavně pro strukturované slovníky pro klasifikaci zdrojů, vizuální navigace strukturou pojmů

**Tezaurus –** Strukturovaný slovník slov. Obsahuje pojmy a relace (hierarchické, ekvivalentní a asociativní).

Používá se pro získávání informačních zdrojů a index pojmů se vztahy mezi nimi.

**Reprezentace znalostí**

**Lexikální část –** specifikuje, které pojmy jsou povoleny ve slovníku reprezentace, např. Defrule: pravidla v jazyce Clips.

**Strukturální část –** specifikuje způsoby, jakými mohou být symboly uspořádány (syntaktická pravidla)

**Procedurální část –** specifikuje přístupové procedury

**Sémantická část –** specifikuje způsoby přiřazení významu popisům

**Schémata reprezentace znalostí**

**Procedurální schémata –** znalosti reprezentované formou procedur, znalosti získáme po provedení procedury. Forma znalosti je implicitní, odpověď na otázku: Jak dojít k řešení?

**Deklarativní schémata –** znalosti reprezentovány formou pravidel. Používá se v umělé inteligenci. Forma znalosti je explicitní, odpověď na otázku: CO platí pro řešení?

**Požadavky na ontologický jazyk –** dobře definovaná syntaxe, sémantika, výkonná odvozovací podpora.

**Sémantická schémata**

Založená na reprezentaci znalostí pomocí ohodnocených grafů. Hlavní reprezentant: Sémantická (asociativní) síť. Lze v ní reprezentovat vztahy: je podmnožinou, je prvkem množiny, pojmy jedinečné, pojmy obecné, taxonomické uspořádání

**Jazyk RDF**

definuje jazyk pro popis zdrojů. RDF zdroj: fyzicky existující objekt. Tvorba metadat různých identifikovatelných zdrojů na webu, obrázek, animace, video. RDF nabízí strojově srozumitelnou sémantiku dat a lepší přesnost při vyhledávání zdrojů než při použití plných textů

**Datový model** - Subjekt - predikát -> subjekt. Subjekt je podmět nebo zdroj. Predikát: název vlastnosti zdroje, Objekt: předmět, hodnota vlastnosti zdroje, který je popisován.

Př. Straka v říši entropie – autor -> Markéta Baňková

RDF je aplikací XML využívající URI. Syntaxe XML zajišťuje pro RDF nezávislost na platformě, rozšiřitelnost, možnost validace, přehlednost.

**URI** – identifikátor pro jednoznačné identifikace zdroje jakéhokoliv obsahu (text, obrázek). Obvykle obsahuje mechanismus pro přístup ke zdroji, adresu počítače, kde je zdroj dostupný a identifikaci zdroje v rámci PC.

Aplikace metadat vyjádřených v RDF: vyhledávání zdrojů, katalogizace, hodnocení obsahu vzhledem k ochraně dětí a soukromí, použití inteligentních agentů pro usnadnění sdílení či výměny znalostí.

**Jazyk RDFS**

Nadstavba jazyk RDF, které je vhodnější pro reprezentaci ontologií. Vznikl rozšířením syntaxe a sémantiky RDF. RDFS nabízí navíc specifikace definičního oboru a oboru hodnot, specifikace datových typů, reprezentace tříd a jejich hierarchií a tvorbu komentářů.

Výhody RDFS: nižší výpočetní složitost, usuzování v RDF je na stejné úrovni jako v RDFS, tvorba jednoduchých ontologických struktur

Nevýhody RDFS: nevhodné pro tvorbu komplexních ontologií, OWL je vhodnější, nelze vyjádřit ekvivalenci tříd, inverzní vlastnosti, omezení kardinality vlastností

**Existenční omezení –** someValuesFrom – pro každého jedince z A musí existovat alespoň jeden jedinec z třídy B, se kterým je ve vztahu X

**Univerzální omezení –** allValuesFrom – vegetariánská pizza může obsahovat jen sýrovou nebo zeleninovou přísadu

**ZT2**

**Vyhledávací stroje** – Shromažďuje informace o webových dokumentech analýzou obsahu webového prostoru pro tvorbu tzv. indexu. Kvalita hledání záleží na kvalitě indexu, způsobu dotazování uživatele a způsob, jakým je web. dokument vytvořen.

**Metavyhledávače –** využití již existujících vyhledávacích strojů, nevytváří indexy, ale hrají prostředníka mezi uživatelem a ostatními vyhledávači. Uživatel má jedno GUI, pomocí kterého zadává dotazy. Výhodou je jen jedno rozhraní, pomocí kterého se dotazuje více vyhledávačů, nevýhodou naopak pomalost kvůli zapojení několika vyhledávacích systémů.

**TopicMaps –** reprezentace informací a znalostí pro lepší nalezení v prostředí webu.

Námětová mapa: dokument reprezentuje soubor námětů a tvrzení o nich

Základní elementy: Námět a téma – cokoliv bez ohledu na to, zda to existuje, ale chceme o tom hovořit

Asociace – vyjadřuje vztah mezi náměty

Výskyt – podává bližší informaci o námětu

**Vrstvy námětové mapy** – znalostní = koncepty, asociace mezi koncepty, vrstva informací = text, obraz, zvuk

**Typ námětu –** vystihuje skupinu prvků se stejnými charakteristikami

**Námět –** představuje konkrétní prvek, o kterém uvádíme jeho vlastnosti a vztahy

**Typ asociace –** vyjadřuje obecný předpis vztahu mezi náměty. Je určen názvem, rolemi a povolenými typy účastníků rolí. Není orientovaná.

**Asociace –** vyjadřuje konkrétní vztah mezi náměty, je to základ sémantické mapy. Arita asociace = kolik námětu v asociaci vystupuje.

**Role –** vyjadřuje způsob účasti aktéra v asociaci

**Typ výskytu –** vyjadřuje obecný předpis vztahu mezi námětem a info o něm, interní (popis) nebo externí (web)

**Výskyt –** popisuje vztah mezi konkrétním námětem a konkrétní informaci o předmětu

**Rozsah platnosti** – Charakteristiky námětů jsou platné vždy v nějakých souvislostech

**Vlastnosti námětu**

**Omezující podmínky –** vymezení formátu hodnot výskytů (datum musí být ve formátu datum), vymezení kardinality asociace/výskytu pro námět (každá osoba má právě jedno datum narození), Vymezení unikátnosti použitých hodnot (email je unikátní)

**Účel a použití námětových map –** Explicitně vyjádřit znalosti tak, že identifikujeme náměty a popíšeme vztahy mezi nimi, propojit informace a znalosti tak, že zaznamenáme, ve kterých zdrojích je možné najít informace o kterých námětech.

Dále slouží k zachycení složitějších hierarchických struktur na základě struktur jednodušších a použitím pravidel.

**Syntaxe XTM (XML TopicMaps)** – vznik syntaxe na základě požadavku využití technologie TopicMaps ve webovém prostředí.

**Syntaxe LTM** – nestandardizovaná syntaxe, oproti XTM je jednodušší, využívá se pro vývoj menších a jednodušších námětových map, může být napsána v jakémkoliv textovém editoru a následně zpracována SW pro námětové mapy.

**Metadata –** data o datech. Příklady: Katalogizační záznam publikace v knihovně, zkrácený výtah, abstrakt.

**Funkce metadat**

shrnutí: sumarizace obsahu

vyhledávání: prohledávání metadat s cílem vyšší přesnosti výsledků

Přístup: zajištění přístupu k datům

Historie: popis historie dat

Metadata pomáhají řešit problémy s slovy s více významy, synonymy a víceznačností.

**Standardizace metadat**

Nutnost použití standardů pro zajištění jednotného způsobu reprezentace metadat. Každé metadatové schéma má omezený počet elementů, pojmenování každého elementu a význam každého elementu.

Problémy mohou být různé projekty pro definici metadat, různé skupiny mohou definovat vlastní způsob tvorby metadat, čímž vzniká problém s kompatibilitou.

**Metadatové systémy**

**Jednoduchá metadata –** obsahují nestrukturovaná data, která jsou automaticky extrahována ze zdrojů a poté indexována pro potřeby vyhledávání

**Strukturovaná metadata –** umožňují dostatečně úplný popis zdroje do té míry, aby mohl každý uživatel posoudit užitečnost zdroje. Jsou tvořeny převážně manuálně, využívají určité standardy.

**Čistá metadata –** zajišťují úplný popis zdroje, využívají se hlavně pro vyhledávání zdroje

**Kódování metadat**

Standard definuje pouze sémantiku. Pro PC je potřeba řešit syntaxi i formální část. V současné době se využívají značkovací jazyky pro k´dování metadat: SGML, HTML, XML, RDF, OWL, TopicMaps.

**Nezbytné vlastnosti pro kódování –** srozumitelnost po PC i člověka, podpora více jazyků, možnost seskupování elementů

**Problémy s metadaty –** Tvorba (tvorba metadat spolu s tvorbou zdroje) a uložení metadat )vnoření metadat do samotného zdroje) – využíváno hlavně u webových dokumentů.

Další problém je s přirozeným jazykem při specifikaci metadat, identifikaci zdrojů (Indetifikátory pro knihy ISBN, ISSN), Identifikátory pro web (URL, URI).

**Generátory metadat -** pro minimalizaci ruční tvorbymetadat. Princip: po zadání URL generátoru se snaží vybrat ze zdroje údaje, které budou součástí metadat

**Slučování námětů** – ke sloučení dojde tehdy, kdy oba dva reprezentují stejný předmět, což poznáme z jejich identifikátorů. Výsledkem je jeden námět, který obsahuje sjednocení všech vlastností původních námětů, duplicity jsou odstraněny

**Slučování námětových map** – princip je stejný jako u námětů

**Validace námětové mapy –** ověřujeme, zda tvrzení v námětové mapě odpovídají omezením pro tato tvrzení.

**TMCL –** obsahuje schéma pro definici struktur, které lze pro tvorbu omezení použít. Využívá se jen pro specifikaci omezení bez možnosti interference

**OSL(Ontopia Schema Languague) –** jazyk pro definici omezení. **Vizualizace –** slouží pro získání prvotní představy o obsahu námětové mapy. Od vizualizačního nástroje požadujeme, aby nabídl zobrazení základní struktury mapy, možnost zaměřit se na specifickou část a možnost volení různých detailů pohledu na elementy námětové mapy.

**Přístupy pro vizualizaci –** Grafy – Statické grafy (celá mapa), dynamické grafy (možnost zobrazení části mapy dle našeho požadavku)

Stromy – hierarchický pohled na data, pro globální přehled

Mapy – jasný a přehledný obraz o informacích v námětové mapě

**Dotazování**

Námětová mapa je abstraktní struktura, která obsahuje informace a znalosti z určité oblasti zájmu. Navigace sama nestačí, a proto potřebujeme dotazovací jazyk. Ten by měl vracet výsledky, přidat informace do námětové mapy a také je odstranit.

**TOLOG**

Jeden z nejúčelnějších dotazovacích jazyků, vychází z programovacího jazyk Prolog a je inspirován SQL. Dotazovat se lze na náměty určitých typů, typy asociací, všechny náměty vystupující v rolích…

Základní konstrukcí je predikát s parametry. Predikát zachycuje vztahy mezi objekty. Parametrem může být proměnná, konstanta nebo řetězcová hodnota. V rámci Tologu rozlišujeme predikáty vestavěné predikáty ( instance-of, záleží na pořadí, jak uvádíme jeho parametry) a uživatelské predikáty (studuje, pracujePro). Predikáty lze skládat pomocí logických spojek:

, a zároveň

| nebo

not negace

**Nikdy neselhávající věta –** někdy chceme získat dotazem i data, která nemusí být v mapě obsažena, třeba proto, že nejsou hodnoty požadovány. Tato věta vrací hodnotu proměnné, ale nevede k chybě, když hodnotu nenajde.

**Projekce –** pokud používáme více proměnných, ale zajímá nás jen některá, použijeme projekci. K tomu složí klíčové slovo Select. Podobné jako v SQL.

**Počítání –** v rámci příkazu select můžeme zjistit počet řádků v tabulce příkazem count.

**Třídění –** výsledky dotazu lze třídit vzestupně nebo sestupně pomocí **asc** nebo **desc**

**Stránkování –** omezí výsledek zobrazení dat jen na omezený počet, je to pomocí příkazu limit „číslo“

**Predikáty porovnání - /=, >=, <** a tak dále

**Odvozovací pravidla –** některé informace nejsou zcela zřejmé a tak lze využít odvozovacích pravidel.

pracujeKde($clovek, $misto):-{pracujeV($clovek : person, $firma : zamestnavatel), (

sidli($firma : zamestnavatel, $misto : sidlo)}.

**Dotazovací systémy –** systému se lze dotazovat v přirozeném jazyku, který je převeden do jazyka Toma. Systém je schopen odpovídat na základní otázky.

**Informační proces**

Informace musím transformovat do vhodné struktury. Přenos informací zprostředkovávají informační systémy.

**Informační zdroj** je obtížně definovatelný. Je to systém, který je reálným nebo potencionálním nositelem, zprostředkovatelem nebo šiřitelem informací. Dělí se z hlediska poskytování informací.

**Hodnocení informačního zdroje –** podle typu informací, kolik záznamů obsahuje, jak daleko do minulosti sahají uchovávané informace, perioda aktualizace, cena, dostupnost.

**Vyhledávání dokumentů -** 1) potřeba po vyhledávání

2) vytipování vhodného informačního zdroje

3) vyhledání dokumentu ve zdrojích

Cílem je získat vhodný počet dokumentů vyhovujících informačnímu požadavku.

**Popis dokumentu –** identifikační popis (jméno autora, rok vydání), charakteristika obsahu dokumentu

**Hledání informací dle obsahu –** obsah dokumentu se dá vyjádřit různě a do výstupu se pak dostávají i dokumenty nevyhovující informačnímu požadavku. Odchylky nejsou zanedbatelné.

**Hodnocení úspěšnosti vyhledání požadovaných informací**

**Koeficient přesnosti – P**

|  |  |  |  |
| --- | --- | --- | --- |
|  | vybráno | nevybráno |  |
| relevantní | A | B | A+B |
| irelevantní | C | D | C+D |
|  | A+C | B+D |  |

**P = A/(A+C)**

**Koeficient úplnosti - R**

|  |  |  |  |
| --- | --- | --- | --- |
|  | vybráno | nevybráno |  |
| relevantní | A | B | A+B |
| irelevantní | C | D | C+D |
|  | A+C | B+D |  |

**R = A/(A+B)**

**Ideální stav –** P = R = 1. Obvykle je mezi P a R vztah nepřímé úměrnosti. Hledání dle obsahu je nepřesné a neúplné

**index a indexování** – vyhledávání = přiřazení relevantních dokumentů vzhledem k dotazu

* položka indexu = každé slovo, které se vyskytne alespoň jednou v souboru www dokumentů
* zkoumá se plný text nebo jen významnější části
* transformace při indexování – vše je převedeno na malá písmena, stopslova jsou vyloučena, manipulace s koncovkami a tvary slov
* ke každé položce indexu je připojen seznam odkazů na www dokumenty + další údaje

**Tezaurus** – hesla, např. popis, homonyma (slova stejně znějící, ale mající jiný význam), synonyma (slovo, které zní jinak, ale má stejný či podobný význam), antonyma(slova opačného významu), generalizace(souhrnný název), specializace(konkrétní název)

**Vyhledávání na internetu**

**Vyhledávací systém –** prostředek pro lokalizaci informací na internetu. Je navržen tak, aby ovládání bylo intuitivní a uživatelsky přívětivé.

**Princip fungování –** vyhledávací systém je tvořen robotem, indexem (invertovaný soubor) a vlastním vyhledávacím strojem

**Prohledávání do šířky –** nové odkazy se přidávají na konec a začíná se opět od startovního dokumentu. Robot zpracuje všechny dokumenty, na které směřují odkazy A => B, C, D. Poté zpracovává B, C a D. Zpracovává webové dokumenty dle úrovní.

**Prohledávání do hloubky –** robot začne pracovat s menší množinou webových dokumentů. Každá URL z nenavštíveného místa je zaznamenána na začátek seznamu. Po tom, co tento dokument zpracuje, je ze seznamu vyřazen. Robot pak postupuje hlouběji vzhledem k počtu úrovní a vnoření odkazů. Pokud narazí na dokument bez odkazů, vrátí se o krok zpět a pokračuje další položkou nenavštívených míst

**Robot –** program sám vykonává určitou činnost. Komunikuje s webovými servery, od kterých získává webové dokumenty. Ty dává k analýze svému vyhledávacímu systému, který nalezené výsledky robotem zařazuje do databáze. Dokumenty jsou díky indexu dohledatelné. Robot prochází všechny weby bez omezení.

Problémem robotů je nekonečná rekurze, kdy některé stránky obsahují CGI skript, který obsahuje odkaz na další CGI skript. Tím se robot dostane na nekonečnou posloupnost čísel, což je nebezpečné u prohledávání do hloubky. Důsledkem je zhroucení robota z důvodu nedostatku paměti.

Robot dokáže zahltit vzdálený WWW server. Řešením je nastavení zpožďovacího intervalu.

Problémy pak představují dokumenty chráněné hesly nebo v jiném formátu než HTML.

**Životní cyklus dokumentu –** vyhledávací robot analyzuje web, dokumenty, vytvoří index z výsledků robota a dokumenty pak vyhledává uživatel.

Výsledkem indexace je index. Index přiřazuje klíčová slova k množinám WWW dokumentů. Vyhledávací stroj si sám určí, ze kterých částí dokumentu vytvoří index. V indexu se zaznamenává datum a čas poslední aktualizace.

Pro zvýšení úspěšnosti ve vyhledávání se slova určitým způsobem transformují (sjednocení velikosti písmen, vyloučení stopslov)

**Stopslova -** slovo, které se v daném [jazyce](http://cs.wikipedia.org/wiki/Jazyk_(lingvistika)) vyskytuje často, ale nenese žádnou významovou informaci, má zpravidla pouze [syntaktický](http://cs.wikipedia.org/wiki/Syntax) význam; typicky se jedná o [spojky](http://cs.wikipedia.org/wiki/Spojka_(slovní_druh)), [předložky](http://cs.wikipedia.org/wiki/Předložka) atp.

**Invertovaný soubor –** datová struktura pro vyhledávání dokumentů splňujících zadanou podmínku