

Přehled znalostních systémů

Kamil Matoušek, Ph.D.

Informační a znalostní systémy

Znalostní systémy

Data -> Informace -> Znalosti (-> Moudrost -> ???)

- Angl.: Knowledge-Based Systems (KBS)
- Programy k rozšiřování a/či dotazování znalostní báze
- Založeny na metodách UI
- Obsahují především znalostní bázi a inferenční mechanismus
- Aplikace: nemůžeme-li přesně popsat cestu, jak nalézt správné řešení

Znalosti

- ... mnoho neuspokojivých definic ...
- „Něco“, co nám umožňuje využívat zkušeností (vlastních i cizích), nacházet řešení různých problémů či efektivně se rozhodovat.
- Jinými slovy, umožňují nám chovat se inteligentně.

Reprezentace znalostí

Rozdělení znalostí podle toho, jakým způsobem je sdělujeme:

1. Znalosti deklarativní

- Obvykle stručnější, vyžaduje však od toho, kdo jej využívá, některé další znalosti. Ty se týkají jak zvyklostí používaných při vlastním deklarativním záznamu znalostí tak i znalostí obecnějších
- Např.: „Zanesený vzduchový filtr u automobilu zvýší jeho spotřebu.“

2. Znalosti procedurální

- Bývají hůře (či složitěji) sdělitelné .
- Např.: „Má-li automobil zvýšenou spotřebu paliva, ověřte, zda nemá zanesený vzduchový filtr.“

3. Znalosti nevedomé (tacitní)

- Např.: jízda na kole, chůze

Požadavky na systém reprezentace znalostí (poznatků)

- vyjádřitelnost poznatků
- použitelnost reprezentace
- logická adekvátnost reprezentačního systému
- heuristická síla, expresivita a přesnost reprezentace
- začlenitelnost do kontextu již reprezentovaných poznatků
- jednoduchost (pohodlnost) vyjádření (zápisu)

Nepravidlové reprezentace znalostí

Produkční pravidla

- IF-THEN, situace-akce (Expertní systémy)

Nepravidlové reprezentace

- Sémantické (asociativní) sítě (Quillian 1968)
- Rámce (Minsky 1975)
- Rozhodovací stromy
- Objekty

a další (matematická logika, konceptuální grafy, ...)

Reprezentace neurčitosti

Zdroje neurčitosti

- Nepřesnost, nekompletnost, nekonzistence dat
- Vágní pojmy
- Nejisté znalosti

Prostředky pro zpracování neurčitosti

- Bayesovský přístup, Bayesovské sítě
- Faktory jistoty
- Dempster-Shaferova teorie
- Fuzzy logika

Dublin Core – popis dokumentů

- První „properties“ **Metadata Workshop, Dublin, Ohio, 1995**
- V současnosti udržuje: **Dublin Core Metadata Initiative**

Property	Definition
Contributor	An entity responsible for making contributions to the content of the resource
Coverage	The extent or scope of the content of the resource
Creator	An entity primarily responsible for making the content of the resource
Format	The physical or digital manifestation of the resource
Date	A date of an event in the lifecycle of the resource
Description	An account of the content of the resource
Identifier	An unambiguous reference to the resource within a given context
Language	A language of the intellectual content of the resource
Publisher	An entity responsible for making the resource available
Relation	A reference to a related resource
Rights	Information about rights held in and over the resource
Source	A Reference to a resource from which the present resource is derived
Subject	A topic of the content of the resource
Title	A name given to the resource
Type	The nature or genre of the content of the resource

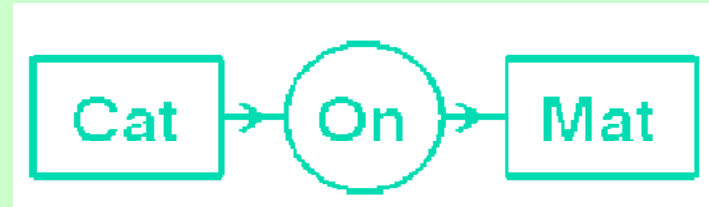
Konceptuální grafy (CG)

- J. Sowa (1976)
 - viz: John F.Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., 2000.
- Logický formalismus zahrnující třídy, relace, individua, kvantifikátory...
- Binarizace relací pomocí reifikace
- Syntaxe
 - Abstraktní syntaxe
 - grafická notace (DF)
 - textová notace (LF)
 - formální jazyk (CGIF)

CG příklady

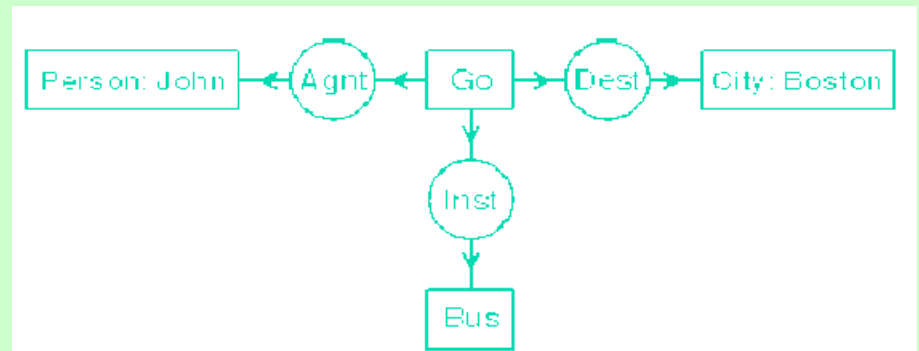
„Acat is on a mat.“

- DF:
- LF:[Cat]->(On)->[Mat].
- CGIF:[Cat: *x] [Mat: *y] (On ?x ?y)



„John is going to Boston by bus.“

- DF:
- LF:[Go]-
 (Agnt)->[Person: John]
 (Dest)->[City: Boston]
 (Inst)->[Bus].
- CGIF: [Go*x] (Agnt?x [Person: John])
 (Dest?x [City: Boston]) (Inst?x [Bus])



Sémantický web

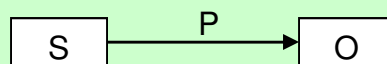
- "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."
 - *Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001*
- Poskytuje společný rámec pro sdílení dat a jejich opětovné využití
- W3C

Resource Description Framework

- jazyk (formát) za účelem sémantické anotace zdrojů
- Samotné RDF – *framework* pro další sémantická rozšíření (RDF Schema, DAML+OIL, OWL, ...), vystihuje pouze základní sémantiku
- Druhy informačních objektů:
 - **zdroj (resource)** má přiřazeno URI
 - **literál (literal)** řetězce, fragmenty textu
- Prvek popisu **triple (statement)** $T = (S, P, O)$
 - *S subjekt* (zdroj), *P predikát* (zdroj), *O objekt* (zdroj).
 - „S má hodnotu vlastnosti P rovnu O“
- RDF model je dán množinou *trojic*

Syntaxe RDF

- **Triples:** výčet všech tvrzení-velmi nepřehledný
- **Orientovaný graf:** každé tvrzení typu (S,P,O)



- **XML** (*serializace*)
 - základní a rozšířená syntax
- **Reifikace** – rozklad
 - (S,P,O) → (rdf:type,X,rdf:Statement)
 - (rdf:Subject,X,S)
 - (rdf:Predicate,X,P)
 - (rdf:Object,X,O)
 - Přejít od tvrzení ke konstruktům vyšších řádů

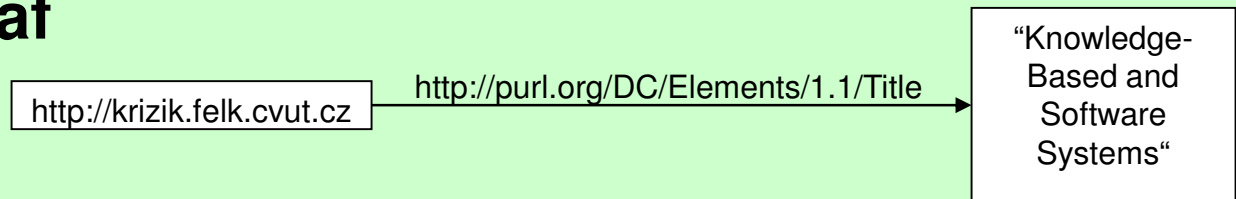
Příklad RDF

„Titulek stránky <http://krizik.felk.cvut.cz> je Knowledge-Based and Software Systems“

- **Triple**

(<http://krizik.felk.cvut.cz>, <http://purl.org/DC/Elements/1.1/Title>, “Knowledge-Based and Software Systems”)

- **Orientovaný graf**



- **XML**

```
<?xml version="1.0" encoding="UTF-8">
```

```
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
  xmlns:dc = "http://purl.org/DC/Elements/1.1/">
```

```
  <rdf:Description about="http://krizik.felk.cvut.cz">
```

```
    <dc:Title> Knowledge-Based and Software Systems </dc:Title>
```

```
  </rdf:Description>
```

```
</rdf:RDF>
```

RDF Schema

Rozšíření syntaxe RDF

- Class třída, koncept
- SubClassOf vlastnost podtřídy
- SubPropertyOf vlastnost podvlastnosti
- Domain vlastnost domény vlastnosti
– definiční obor
- Range vlastnost oboru hodnot
vlastnosti
- Literal koncept literálu

Ontologie

Ontology is an formal, explicit specification of a shared conceptualization.

Tom R. Gruber '93

Ontology is a formal explicit description of concepts in a domain of discourse

Konceptuální modelování

- Tvorba doménových ontologií experty v dané oblasti s minimální znalostí modelování

Softwarové řešení - Protégé

The screenshot displays the Protégé 3.0 interface for editing a class. The window title is "newspaper Protégé 3.0 (file:\C:\Program%20Files\Protege_3.0\examples\newspaper\newspaper.pprj...". The menu bar includes File, Edit, Project, Window, and Help. The toolbar contains various icons for file operations and editing. The main interface is divided into several panes:

- CLASS BROWSER:** Shows the class hierarchy for the project "newspaper". The hierarchy includes:
 - :THING
 - :SYSTEM-CLASS
 - Author
 - News_Service
 - Columnist
 - Editor** (selected)
 - Reporter
 - Content
 - Layout_info
 - Library
 - Newspaper
 - Organization
 - Person
- Superclasses:** Lists "Author" and "Employee" as superclasses of the selected class.
- CLASS EDITOR:** Shows the configuration for the "Editor" class (instance of :STANDARD-CLASS).
 - Name:** Editor
 - Documentation:** Editors are responsible for the content of sections.
 - Role:** Concrete
 - Template Slots:** A table listing slots for the class.

Name	Cardinality	Type
current_job_title	single	String
date_hired	single	String
name	single	String
other_information	single	String
phone_number	single	String
responsible_for	multiple	Instance of Employee
salary	single	Float
sections	multiple	Instance of Section

Deskripční logika

- Vlastně logiky-rodina formalismů, jejichž jádrem je vymezení konceptů pomocí jejich formálních popisů (deskripcí)
- Na základě popisů zjišťována subsumpce konceptů
- Též terminologická logika

Topic Maps

- `/* ONTOLOGY */`
- `/* topic types */`
- `[composer = "Composer" @"http://psi.ontopia.net/music/#composer"]`
- `[opera = "Opera" @"http://psi.ontopia.net/music/#opera"]`
- `/* association types */`
- `[composed-by = "Composed by" = "Composed" /composer
@"http://psi.ontopia.net/music/#composed-by"]`
- `/* occurrence types */`
- `[birth-date = "Birth date" @"http://psi.ontopia.net/biography/#date-of-birth"]`
- `[death-date = "Date of death" @"http://psi.ontopia.net/biography/#date-of-death"]`
- `/* INSTANCES */`
- `/* composer */`
- `[smetana : composer = "Smetana, Bedrich"] {smetana, birth-date, [[...]]} {smetana, death-date,
[[...]]}`
- `[dvorak : composer = "Dvorak, Antonin"] {dvorak, birth-date, [[...]]} {dvorak, death-date, [[...]]}`
- `[janacek : composer = "Janacek, Leos" @"http://psi.ontopia.net/composer/janacek"] {janacek,
birth-date, [[1854-07]]} {janacek, death-date, [[1928-08]]}`
- `/* opera */ /* by Smetana */`
- `[bartered-brid : opera = "The Bartered Bride"] /* by Dvorak */`
- `[rusalka : opera = "Rusalka"] /* by Janacek */`
- `[sarka : opera = "Sarka"] [beginning-of-a-romance : opera = "The Beginning of a Romance"]`
- `[jenufa : opera = "Jenufa" @"http://psi.ontopia.net/opera/jenufa"]`
- `[osud : opera = "Osud"]`
- `/* composed by associations */`
- `composed-by(bartered-brid : opera, smetana : composer)`
- `composed-by(rusalka : opera, dvorak : composer)`
- `composed-by(sarka : opera, janacek : composer)`
- `composed-by(beginning-of-a-romance : opera, janacek : composer)`
- `composed-by(jenufa : opera, janacek : composer)`
- `composed-by(osud : opera, janacek : composer)`

Softwarové řešení - Omnigator

[Omnigator] Welcome Page - Microsoft Internet Explorer

Soubor Úpravy Zobrazit Oblíbené Nástroje nápověda

Adresa <http://www.ontopia.net/omnigator/models/index.jsp>

omnigator eight

The free Seamless Knowledge browser from Ontopia. Powered by the Ontopia Knowledge Suite.

[Manage](#) | [Customise](#)

Welcome to the Omnigator

Index of Topic Maps

- concert.rdf
- factbook.hytm
- i18n.ltm
- jill.xtm
- mondial.xtm
- opera.hytm
- opera.xtm
- support-kb.ltm
- tm-standards.xtm
- xmltools-tm.xtm

Version

Version 2.1.1
Build 2005-01-13 #1522
Expires No expiry

Further Information

- Documentation
 - Omnigator User Guide
 - Query Language Tutorial
 - Schema Language Tutorial
- Articles
 - The TAO of Topic Maps
 - The XML Papers
 - Living with Topic Maps and RDF
 - Metadata? Thesauri? Taxonomies? Topic Maps!

Read Me First

The Omnigator is a technology showcase and teaching aid designed to demonstrate the power of Topic Maps. It is also used extensively as a topic map debugger and prototyping tool. Now, with the introduction of RDF support, the Omnigator is evolving into a multi-purpose Semantic Web Agent.

Features

The Omnigator is omnivorous: It eats anything, provided it is a topic map – or can be viewed as one! Its most unique feature is that it lets you load and navigate any conforming topic map, whether its format be XTM, HyTM, LTM – or even RDF. What's more, although it isn't optimized for any particular application, the Omnigator is able to make reasonable sense out of any reasonably sensible topic map!

But the Omnigator lets you do far more than simply browse your topic maps. It supports all aspects of the Topic Maps standard and has a host of powerful features, such as the ability to merge topic maps on the fly; search in ways that make Google boggle; export to a range of syntaxes; customize different views; produce filtered subsets based on scope; perform semantic validation; and much more besides.

http://www.ontopia.net/omnigator/plugins/viz/viz.jsp?tm=opera.hytm&id=puccini&redirect=%2Fomnigator%2Fmodels%2Ftopic_complete.j

Internet

Znalostní inženýrství

- Oblast umělé inteligence, která vyvíjí znalostní systémy. Tj. počítačové programy které obsahují velké množství znalostí, pravidel a usuzovací mechanismy dávající řešení problémů reálného světa.
- Častým typem znalostního systému je expertní systém, napodobující usuzovací procesy zkušeného odborníka - experta (tj. člověka, který provádí svou profesi mnoho let). Typickými příklady expertních systémů jsou diagnostika bakteriálních infekcí, poradenství při průzkumu nerostů a posuzování návrhů elektrických obvodů.

Principy znalostního inženýrství

Od poloviny 80. let 20. stol. mnoho principů, metod a nástrojů výrazně zlepšilo proces získávání znalostí.

- Existují různé typy znalostí a že pro požadované znalosti je třeba volit správný přístup a metodu.
- Existují různé typy expertů a odborných znalostí, takže je třeba vybrat tomu odpovídající vhodné metody.
- Existují různé způsoby reprezentace znalostí, které mohou pomoci při získávání, validaci a znovuvyužití znalostí.
- Existují různé způsoby použití znalostí - proces získávání může být veden cíli projektu.
- Jsou používány strukturované metody ke zvýšení efektivity procesu získávání znalostí.

Knowledge Management

- = strategie nebo systém vyvinutý aby pomohl organizacím vytvářet, zachycovat, analyzovat, aplikovat, a znovuvyužívat znalostí, aby dosáhly konkurenční výhody.
- Klíčové v KM:
Se znalostmi v organizaci se nakládá jako s klíčovým bohatstvím!
- Jednoduchá fráze shrnující základní aspekt KM je *dát správné znalosti správným lidem ve správný čas ve správné podobě.*

Metody KM:

- podpora **předávání** znalosti v organizaci
 - Metody komunikace tváří v tvář, např. *peer assist, lessons learnt reviews, knowledge fairs*
 - Počítačové komunikační metody, např. email, Lotus Notes, komunity zkušeností
 - Ukládání a vyhledávání v počítačových systémech, např. intranet, knihy znalostí
 - Znalostní systémy, např. expertní systémy
- podpora **tvorby** nových znalostí

Knowledge Acquisition (KA)

= **získávání (elicitace), shromažďování, analýza, modelování a validace znalostí** ve znalostním inženýrství a managementu

<http://www.epistemics.co.uk>

Důležitost získávání znalostí

- Získávání dostatečného množství kvalitních znalostí jako základu pro užitečný robustní systém bylo po mnoho let zdoluhavé a drahé a bylo také nejkritičtějším místem při budování expertních systémů. Proto se stalo velkou výzkumnou oblastí znalostního inženýrství.
- Cílem získávání znalostí je vyvinout metody a nástroje, které obtížnou úlohu zachycení a validace znalostí experta umožní provést maximálně účinně a efektivně. Experti bývají důležitými a zaměstnanými lidmi a proto je nezbytné, aby používané metody minimalizovaly čas, který experti stráví mimo práci účastí na sezeních při získávání znalostí.

Problémy KA

- Většina znalostí leží v hlavách expertů
- Experti mají ohromné množství znalostí
- Experti mají mnoho „tacitních“ znalostí
 - Nevědí vše, co znají a používají
 - Tacit znalost je těžké (nelze) popsat
- Experti jsou velmi zaneprázdnění a cenní lidé
- Žádný expert neví vše (potřebné)
- Znalosti mají „trvanlivost“

Požadavky na metody KA

- Vztít experty na kratší chvílky mimo práci
- Umožnit i neexpertům pochopit znalosti
- Zaměřit se na základní znalosti
- Schopnost zachytit tacitní znalosti
- Dovolit shromažďování znalostí od různých expertů
- Musí umožnit validaci a správu znalostí

Metody KA (1)

- Metody tvorby protokolu
 - různé typy interview (nestrukturovaná, částečně strukturovaná a strukturovaná), metody podávání zpráv (jako *self-report* a *shadowing*) a metody pozorování
- Metody analýzy protokolu
 - používá se s přepisy z interview či jiné textové informace k určení různých typů znalostí, jako cíle, rozhodnutí, vztahy a atributy. Je mostem mezi použitím metod založených na protokolu a metod modelování znalostí.
- Metody tvorby hierarchie
 - např. *laddering*, pro stavbu taxonomií či jiných hierarchických struktur, jako jsou stromy cílů a rozhodovací sítě.
- Maticové metody
 - zahrnují konstrukci tabulek ukazujících věci jako vyskytující se problémy a možná řešení. Důležitými typy jsou **rámce** pro reprezentaci vlastností konceptů a metoda **repertoárové tabulky** pro získání, ohodnocení, analýzu a kategorizaci vlastností konceptů

Metody KA (2)

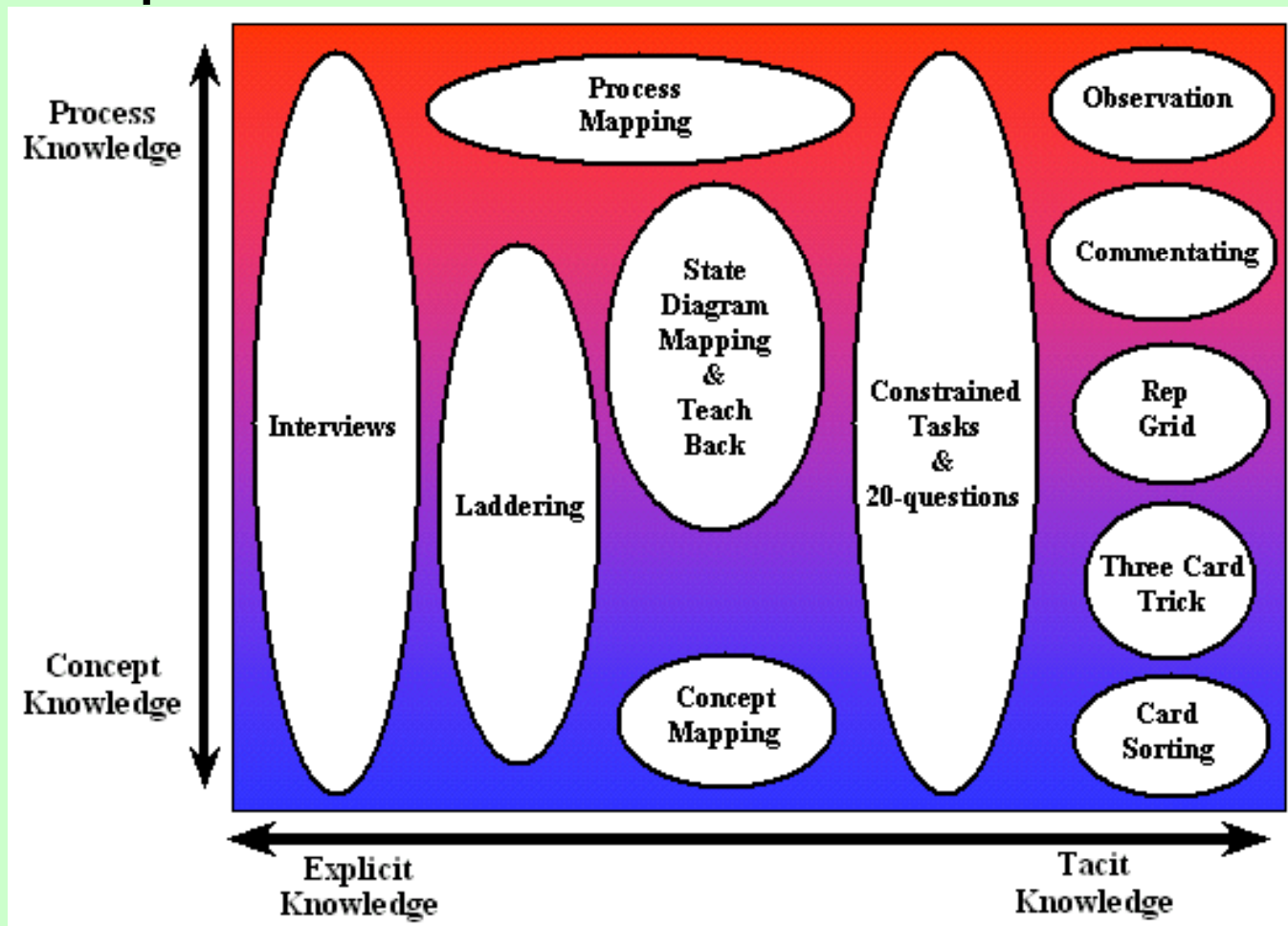
- Metody třídění
 - zachycení způsobu, kterým lidé porovnávají a třídí koncepty
 - mohou vést k odhalení znalostí o třídách, vlastnostech a prioritách.
- Metody omezování informací a zpracování
 - omezit čas a/nebo informace experta při provádění úloh. Např. *metoda dvaceti otázek* je efektivním způsobem, jak se dostat ke klíčovým informacím z oboru (domény) s uvedením priorit.
- Metody založené na diagramech
 - obsahují tvorbu a užití map konceptů, sítě stavů a přechodů, diagramy událostí a mapy procesů. Jejich použití je zvláště důležité pro odpovědi na otázky „*co, jak, kdy, kdo a proč*“, které se týkají úloh a událostí.

Hypotéza rozdílného přístupu

- Proč mít tolik metod?
- Faktem je, že je mnoho různých typů znalostí nošených experty a různé metody jsou potřeba pro „zpřístupnění“ různých typů znalostí.
- Tato hypotéza se označuje „differential access hypothesis“, a podporují ji experimentálně získané důkazy.

Srovnání metod KA

- Procesní vs. objektové znalosti
- Explicitní vs. tacitní znalosti



Typické použití metod KA (1)

Jednoduchý způsob použití popsaných metod (od běžných k méně přirozeným):

1. Počáteční rozhovor s expertem (zaznamenat všechna sezení)
 - a) zjištění rozsahu, jaké znalosti je třeba získat,
 - b) určení, k jakému účelu má být znalost vložena,
 - c) porozumění alespoň částečné klíčovým termínům a
 - d) vytvoření si vztahu k expertovi.
2. Přepis počátečního rozhovoru a analýza výsledného protokolu
 - a) žebříček konceptů výsledných znalostí z hlediska široké reprezentace znalostí v dané oblasti
 - b) množina otázek pokrývajících základní problémy v celé doméně
3. Částečně strukturovaný rozhovor s expertem za použití připravených otázek
 - získání struktury a zaměření v oblasti
4. Přepis a analýza částečně strukturovaného rozhovoru
 - Výsledný protokol: typy znalostí, které se vyskytují.
 - Typicky: koncepty, atributy, hodnoty, vztahy, úlohy a pravidla.
5. Reprezentace částí znalostí nejvhodnějšími modely
 - Žebříčky, tabulky, síťové diagramy, hypertext, atd.
 - Též strukturovaně vtipy, obrázky a vysvětlení – hypertext a různé nadpisy

Typické použití metod KA (2)

6. Použití „nepřirozených“ metod na výsledné modely a strukturovaný text
 - *laddering*, řešení problémů s „myšlením nahlas“, metoda dvaceti otázek, repertoárová tabulka
 - umožňuje expertovi změnit a rozšířit již zaznamenané znalosti
 7. Opakování analýzy, tvorby modelu a získávacích sezení
 - dokud expert a znalostní inženýr nejsou spokojení s výsledkem
 8. Validace dalšími experty, potřebné změny znalostí
- Navíc: dříve získané znalosti, obecně použitelné znalosti
 - Cílem bývá znovu použít co nejvíce dříve získaných znalostí.
 - Podpůrné metody (ontologie a modely řešení problémů) poskytují znalosti obecně použitelné k nadhození nápadů expertovi, jako např. obecné třídy objektů v doméně a obecné způsoby provádění úloh.
 - „Re-use“ znalostí – základ nejefektivnějšího získávání znalostí.
 - Čím více znalostí (nadhled: obecné znalosti), tím je proces účinnější.
 - Často kombinace přístupů „založeného na teorii“ (shora-dolů) a „řízeného daty“ (zdola-nahoru).

Novější směry v KA (1)

- **Metodologie** – projekty znalostního inženýrství řízeny modely
 - „kostry“ + obecné znalosti, pomáhají při získávání znalostí
 - CommonKADS: Modely pro *vedení projektu: organisation model, task model, agent model, expertise model, communications model a design model.*
 - CommonKADS: Modely *řešení problémů* pomáhají při tvorbě nových modelů, popisují role znalostí v úlohách a radí, na které typy znalostí se zaměřit.
 - CommonKADS: Spirálový přístup, fáze *přezkoumávání, stanovování rizika, plánování a sledování* - umožňuje rychlý vývoj systému.
- **Ontologie** – formalizované reprezentace znalostí domény z určitého úhlu pohledu.
 - Sdílení a předávání znalostí mezi lidmi a mezi počítačovými systémy.
 - Existují mnohé obecné ontologie použitelné v různých oblastech znovuvyužitím obsažených znalostí.
 - Projekt může začít s mnoha rámcovými kostrami, které jsou předem definovanými strukturami pro zaznamenávané znalosti.
 - Jako u modelů řešení problémů v CommonKADS, ontologie také vedou znalostního inženýra k určení typů znalostí, které je třeba prozkoumat.

Novější směry v KA (2)

- **SW nástroje** v procesu KA.
 - SW balíčky, jako PCPACK, obsahují nástroje pro analýzu, strukturování a uložení požadovaných znalostí.
 - Nástroje k modelování a centrální databáze znalostí mohou poskytovat různé názorné pohledy na oblast zájmu.
 - Také mohou přinutit k dobré disciplíně uživatele, takže i nováčci ve znalostním inženýrství mohou být instruováni při vedení projektů získávání znalostí.
 - SW systémy pro ukládání a indexování mohou též zprostředkovat opakované využití a přenos znalostí mezi projekty.
 - V poslední době jsou vyvíjeny systémy, které používají obecné ontologie k automatické analýze a strukturování znalostí.
- Principy a metody znalostního inženýrství mimo expertní systémy
 - **Knowledge management** (KM) v rámci organizace je strategie, kdy znalosti v organizaci jsou klíčovým bohatstvím, které je třeba spravovat co nejefektivněji.
 - Mnoho principů a metod znalostního inženýrství úspěšně přeneseně pomáhá např. i při tvorbě webových stránek pro intranety společností.

Odkazy

W3C

- <http://w3.org>
- <http://www.w3.org/2001/sw> - Semantic Web Activity
- <http://www.w3.org/2004/OWL> - Web Ontology Language

Konceptuální grafy

- <http://users.bestweb.net/~sowa/cg/>

Tutoriál RDF

- <http://www.w3schools.com/rdf/default.asp>

RDF validátor online

- <http://www.w3.org/RDF/Validator/>

Tutoriál – ontologie

- <http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html>

Introduction to OWL

- http://www.w3schools.com/rdf/rdf_owl.asp

Protege

- <http://protege.stanford.edu>

Topic Maps

- <http://www.ontopia.net>