Podatkovni modeli in jeziki

2018/19

Iztok Savnik, FAMNIT

Osnovni podatki

Naslov:	Podatkovni modeli in jeziki
Predavatelj:	dr. Iztok Savnik
Vaje:	domače naloge
Točke:	6 KT
Komunikacija:	e-učilnica, forumi, e-posta, govorilne ure
Govorilne ure:	po predavanju
URL:	osebje.famnit.upr.si/~savnik/predmeti/PMJ
E-učilnica:	e.famnit.upr.si

Ocenjevanje

- Sestava ocene:
  - Ocena domačih nalog – 40%
  - Ocena pisnega izpita – 60%

- Vsaka od delnih ocen mora biti pozitivna!

Domače naloge

1) Domača naloga po večjem sklopu snovi
   - 4-5 domačih nalog
   - Manjši primeri uporabe podatkovnih modelov in jezikov
   - Primer: definiraj podatkovno bazo knjižnice z XML, DTD, SSL, XQL, ...
   - Osnova za pisni izpit !
2) Domače naloge = 1 seminar

Vsebina

1) Tekstovne podatkovne baze
2) Model ključ-vrednost
3) Grafovski podatkovni model
4) Baze znanja
5) Objektno-relacijski podatkovni model
6) Logični podatkovni model

Smoter (1)

Poglabljanje znanja

- Razširiti pogled
- Podati izvore
- Predstaviti formalne osnove
- Predstaviti moderne implementacije
- Predstaviti sisteme
- Globalen pogled na jezike in modele

Smoter (2)

- Prikaz izvorov in formalnega ozadja
  - Logika, izjavni račun, predikatni račun
  - Relacijski račun, relacijska algebra
  - Regularni jeziki
  - Grafi, konceptualne mreže
  - Predikatni račun
  - Opisna logika
  - Hornovi stavki

Smoter (3)

Pregled modernih jezikov in modelov

- XML, Xquery
- RDF, SparQL, Linked Data
- OWL, CycL
- RDF3X, Virtuoso, Cstore, Cassandra
- SQL3, PL/SQL
- Prolog, Datalog
- F-Logic, Flora

Smoter (4)

Reference na moderne sisteme

- Virtuoso
  - “innovative enterprise grade multi-model data server for agile enterprises & individuals“
  - Open source: VOS
- Triple-stores
  - 3store, 4store, jena, sesame, rdf3x, algerograph, ...
- Cyc
- Deduktivne podatkovne baze
  - Datalog, F-Logic, Flora
- Načrtovalska orodja
  - Oracle, Informix, ...

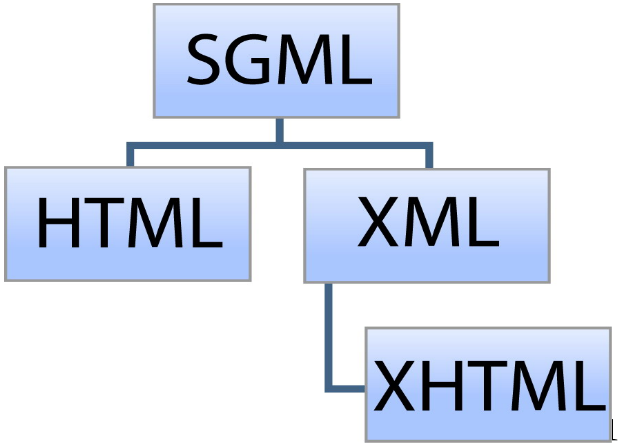
Smoter (5)

Praktično znanje

- Razvoj aplikacij, ki slonijo na uporabi podatkovnih modelov.
- Uporabo modernih orodij in tehnik za implementacijo podatkovnih okolij.
- Razvijanje zmožnosti obvladovanja kompleksnih podatkovnih okolij.
- Razvijanje zmožnosti za razvoj aplikacij v praksi.

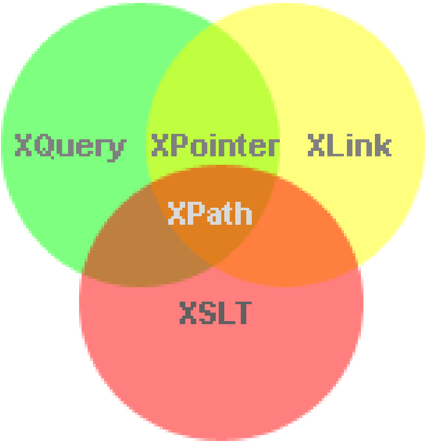
Tekst (1)

- Regularni jeziki
  - Primerni za opis teksta
  - Bližje jezikom
  - BNF, Kontekstno neodvisni jeziki
  - Delno-strukturirani podatki
- Strukturiran tekst
  - XML, SSL
  - XPath, XQL
  - XML-Schema
  - XQuery



Tekst (2)

- Podatkovni model
  - XML, SSL
- Povpraševalni jeziki
  - XQL, Xquery, XML-QL
- Konceptualni modeli
  - XML-Shema
- XML podatkovne baze
  - Virtuoso
  - RDBMS vsebujejo vmesnike do XML



Key-value data model

- NoSQL movement
- Key-Value stores
- Column-oriented database systems
- Map-Reduce systems
- Graph database systems
- Dataflow systems

Motives of NoSQL practioners

- Avoidance of Unneeded Complexity
- High Throughput
- Horizontal Scalability and Running on Commodity Hardware
- Avoidance of Expensive Object-Relational Mapping
- Complexity and Cost of Setting up Database Clusters
- Current “One size fit’s it all” DBMS Thinking Was and Is Wrong
- Myth of Effortless Distribution and Partitioning of Centralized Data Models
- Movements in Programming Languages and Development Frameworks
- Requirements of Cloud Computing
- The RDBMS plus Caching-Layer Pattern/Workaround vs. Systems Built from Scratch with Scalability in Mind
- Yesterday’s vs. Today’s Needs

Background work

- Michael Stonebraker: “The End of an Architectural Era”
  - “that the current RDBMS code lines, while attempting to be a “one size fits all” solution, in fact, excel at nothing”
  - prototype named H-Store developed at the M.I.T. beats up RDBMSs by nearly two orders of magnitude in the TPC-C benchmark
  - RDBMSs“ are 25 year old legacy code lines that should be retired in favor of a collection of “from scratch” specialized engines. The DBMS vendors (and the research community) should start with a clean sheet of paper and design systems for tomorrow’s requirements, not continue to push code lines and architectures designed for yesterday’s needs”
  - “popular relational DBMSs all trace their roots to System R from the 1970s”: IBM’s DB2 is a direct descendant of System R, Microsoft’s SQL Server has evolved from Sybase System 5 (another direct System R descendant) and Oracle implemented System R’s user interface in its first release.

Design Considerations

- Main Memory
- Multi-Threading and Resource Control
- Grid Computing and Fork-Lift Upgrades
- High Availability
  - “start with shared-nothing support at the bottom of the system”
- No Knobs
  - current RDBMSs were designed in an “era, [when] computers were expensive and people were cheap. Today we have the reverse.

Stonebraker's Critical Reception of NoSQL Databases

- "The "NoSQL" Discussion has Nothing to Do With SQL"
- "a low-level record-at-a-time DBMS interface, instead of SQL"
- Two reasons for moving towards non-relational datastores:
  - flexibility and
  - Performance
- Five sources of performance overhead:
  - Communication
  - Logging
  - Locking
  - Latching
  - Buffer Management
- NoSQL databases also have to address the components of performance overhead mentioned above
  - "I fully expect very high speed, open-source SQL engines in the near future that provide automatic sharding. [. . .] Moreover, they will continue to provide ACID transactions along with the increased programmer productivity, lower maintenance, and better data independence afforded by SQL."

Logika

- Formalna osnova modelov in jezikov
- Izjavni račun
  - Osnovne metode
- Predikatni račun
  - Relacijski račun
  - Opisna logika = strukturiran del PR
  - Prolog = del PR (hornovi stavki)
  - Semantični modeli
- Metode
  - Sklepanje, Dedukcija
  - Resolucija

Graph database systems

- Graph data model
  - Data is represented in the form of the graph
  - Any representation can be converted to a graph representation
- Graph representations
  - Adjacency lists
  - Adjecency matrix
  - Triples and triple tables
  - Special data structures
    - Indexes, bitmaps, signature trees, ...
- RDF data model
  - Many levels of representation: data, schema, logic

Graph data model

- Graph database
  - Database that uses graphs for the representation of data and queries
- Vertexes
  - Represent things, persons, concepts, classes, ...
- Arcs
  - Represent properties, relationships, associations, ...
  - Arcs have labels !

RDF

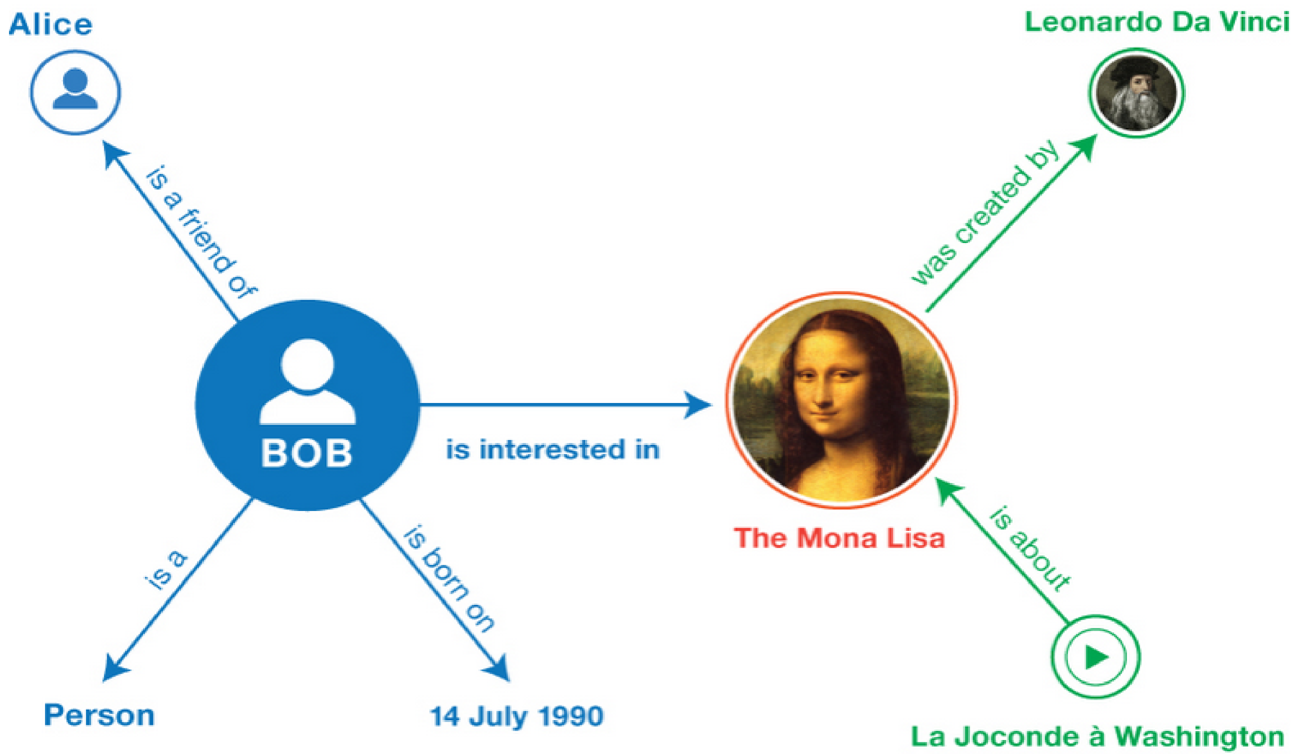
- Resource Description Framework
  - Tim Berners Lee, 1998, 2009 ...
  - This is movement !
- What is behind ?
  - Graphs are fundamental representation ?
    - Can express any other data model
    - Many times serve as the theoretical basis
  - Graphs can represent data and knowledge ?
    - Data and knowledge will be integrated in novel applications
    - Many reasoners use triple-representation of knowledge and data, e.g., Cyc

RDF

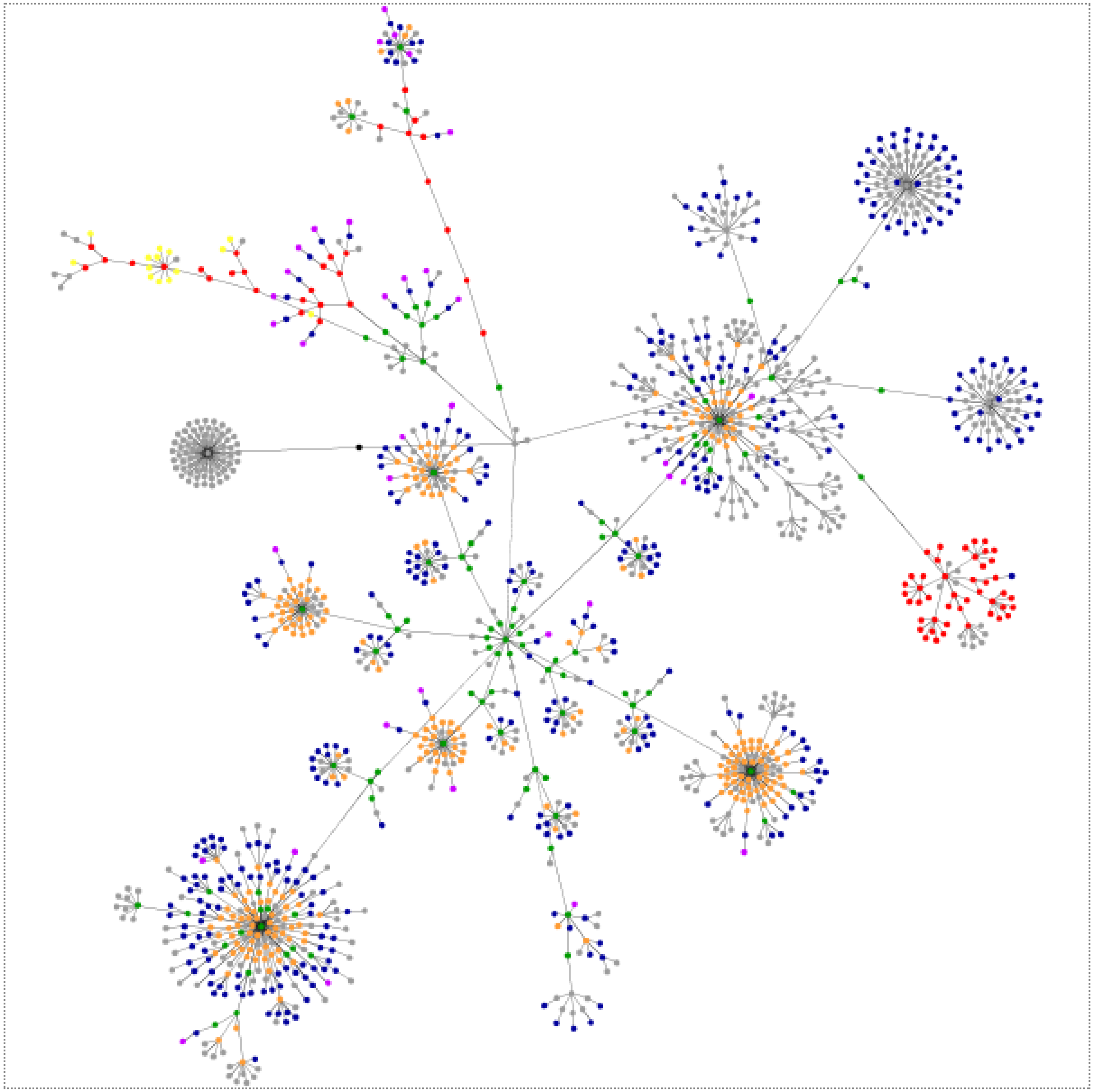
- Novel applications require some form of reasoning
  - Intelligent assistants, system diagnostics, ...

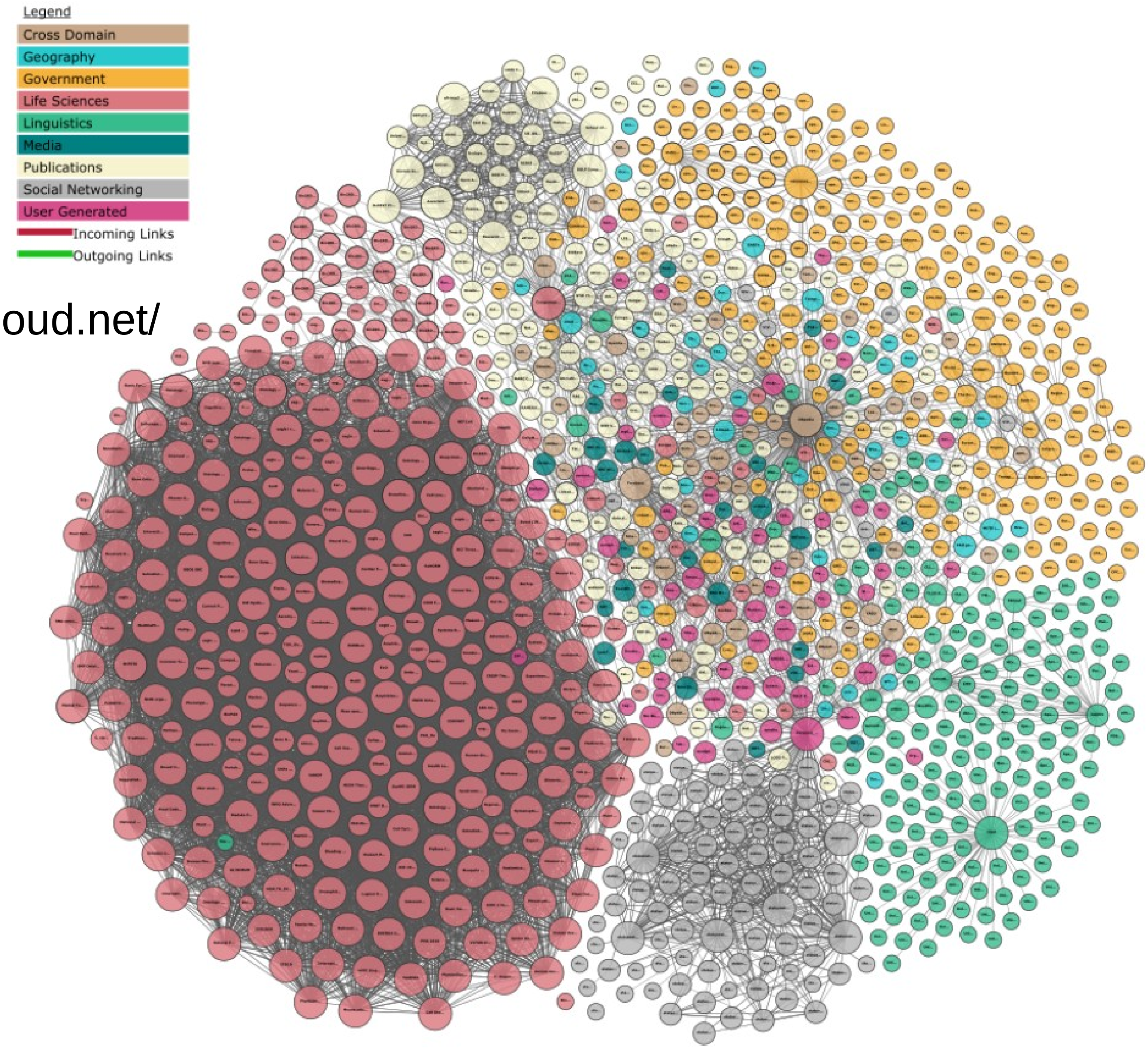
RDF

```
<Bob> <is a> <person>.
<Bob> <is a friend of> <Alice>.
<Bob> <is born on> <the 4th of July 1990>.
<Bob> <is interested in> <the Mona Lisa>.
<the Mona Lisa> <was created by> <Leonardo da Vinci>.
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>
```



Vizualizacija RDF baz





http://lod-cloud.net/

RDF Schema

- RDFS
- Knowledge representation language
  - Not just graph any more !
  - AI Frames, Object Model
- Small dictionary for RDFS
  - rdfs:class, rdfs:subClassOf, rdfs:type
  - rdfs:property, rdfs:subPropertyOf
  - rdfs:domain, rdfs:range

Graph database systems

- Querying
  - Initially in-memory systems
  - SPARQL query language
    - Data and knowledge query language (RDF inference)
  - Heavy use of indexing
    - Special new index structures
  - Query optimization
    - Dynamic programming, pipelines, bushy trees
  - Distributed databases and query processing

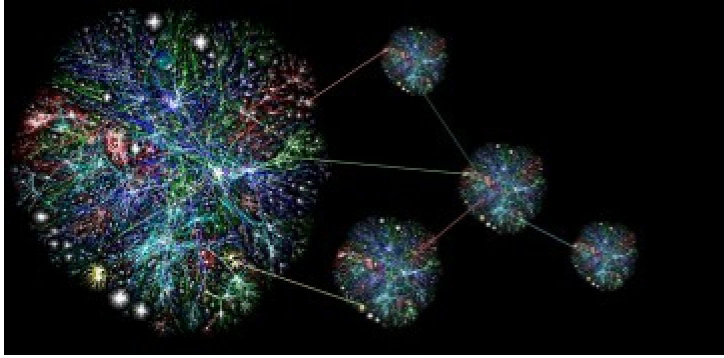
SPARQL

- SPARQL Protocol and RDF Query Language
- SPARQL query
  - Graph can include variables in place of constants
- Operations
  - JOIN (natural, left-join)
  - AND, FILTER, UNION, OPTIONAL
- Commercial DBMS-s
  - Implement RDF and SPARQL

Baze znanja

- Logika !
  - Konceptualni grafi (john F. Sowa, 1981)
  - Terminologije -> Ontologije
- Jeziki za predstavitev znanja
  - KL-ONE, Classics, Back
  - Sklepanje
- Opisna logika (description logic)
  - Grafi kot logika + Grafi kot podatki
  - RDF, RDFS, OWL, Rules

Baze znanja



- Opisna logika
  - Strukturiran del PR
  - Lahko predstavimo s trojicami: v osnovi imamo RDF
  - XML zapis
- OWL
  - W3 standard
  - Opisna logika razširjena s praktičnimi jezikovnimi gradniki
- Sklepanje
  - Povpraševalni jezik
  - Sklepanje s koncepti
  - Sklepanje na osnovi konceptov in podatkov

Opisna logika

- Zadnje ime za družino jezikov za predstavitev znanja
- Način predstavitve znanja
  - Predstavitev osnovnih konceptov domene
  - Uporaba konceptov za predstavitev dejstev (logičnih zvez)
- Za razliko od predhodnikov je predstavitev znanja osnovana na logiki
  - Semantika jezika je vezana na logiko
  - Omogočeno je sklepanje: izpeljava znanja, ki je implicitno iz eksplicitno zapisanega znanja

Strukturirana omrežja dedovanja

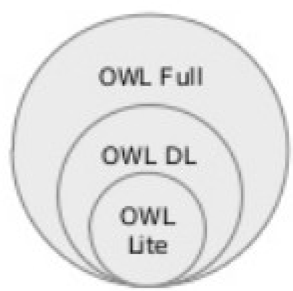
- DL je naslednik strukturiranih omrežij dedovanja [Brachman, 1977b; 1978]
  - Razrešitev dvoumnosti, ki so prisotne v semantičnih mrežah in okvirjih
- Prva realizacija: KL-One [Brachman and Schmolze, 1985] vsebuje naslednje 3 ideje:
  - Osnovni sitaktični objekti: koncepti, vloge in konstante (unarni, binarni predikati in konstante v PR)
  - Majhna množica konstruktorjev za gradnjo kompleksnih konceptov in vlog iz enostavnih
  - Implicitno znanje lahko izpeljemo z uporabo sklepanja
    - Vsebovanje konceptov (vlog) in relacija instanciranja
    - NE isa relacija kot v semantičnih mrežah!
    - Pomembne so relacije def na osnovi lastnosti

OWL and Description Logics

OWL exists in three variants with a different degree of expressivity:

OWL Lite
- classification hierarchy
- simple constraints
- cardinality constraints
- cardinality values [0..1]
- no container classes
- no disjunction
- simple computable and decidable inferences
- complexity: ExpTime (worst case)

OWL DL
- based on DL
- includes all OWL language constructs
- several limitations, e.g., no reification
- computable and decidable inferences
- complexity: NExpTime (worst case)

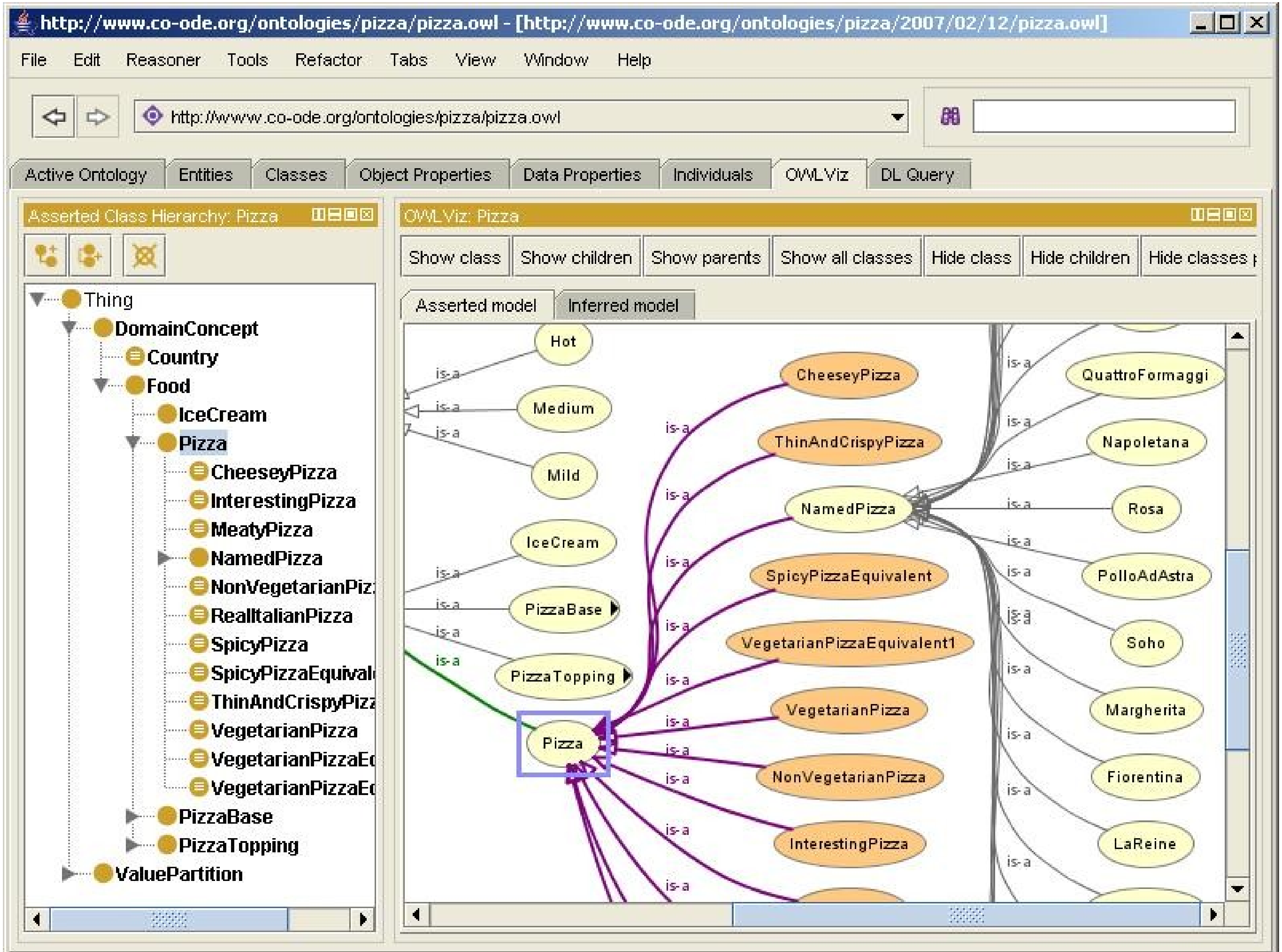


OWL Full
- highest level of expressivity
- a class can be treated simultaneously as a collection of individuals, and as an individual in its own right
- allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary.
- complex reasoning allowed but no guarantee to be computable and decidable

OWL konstrukti

OWL konstrukt	DL	Primer
intersectionOf	$C_1 \sqcap \ ... \ \sqcap C_n$	Human⊓Male
unionOf	$C_1 \sqcup \ ... \ \sqcup \ Cn$	Doctor⊔Lawyer
complementOf	$\neg C$	¬Male
oneOf	$\{o_1, ..., o_n\}$	{john, mary}
allValuesFrom	$\forall P.C$	∀hasChild.Doctor
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer
value	$\exists P.\{o\}$	∃citizenOf .USA
minCardinality	$\geq nP.C$	≥2hasChild.Lawyer
maxCardinality	$\leq nP.C$	≤1hasChild.Male
cardinality	$=nP.C$	=1hasParent.Female

+ XML Schema tipi: int, string, real, etc...

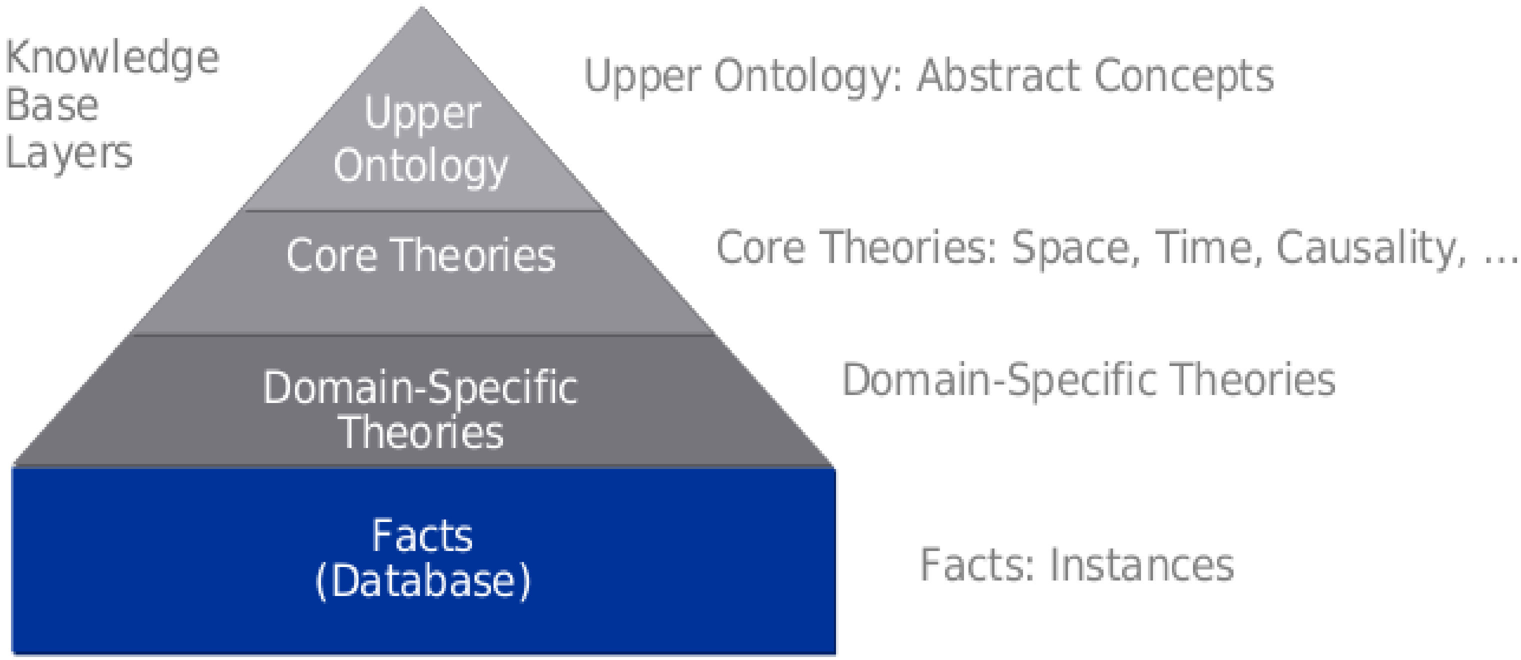


http://www.co-ode.org/ontologies/pizza/pizza.owl - [http://www.co-ode.org/ontologies/pizza/2007/02/12/pizza.owl]
File
Edit
Reasoner
Tools
Refactor
Tabs
View
Window
Help
http://www.co-ode.org/ontologies/pizza/pizza.owl
Active Ontology
Entities
Classes
Object Properties
Data Properties
Individuals
OWLViz
DL Query
Asserted Class Hierarchy: Pizza
Thing
DomainConcept
Country
Food
IceCream
Pizza
CheeseyPizza
InterestingPizza
MeatyPizza
NamedPizza
NonVegetarianPiz
RealItalianPizza
SpicyPizza
SpicyPizzaEquival
ThinAndCrispyPizz
VegetarianPizza
VegetarianPizzaEq
VegetarianPizzaEq
PizzaBase
PizzaTopping
ValuePartition
OWLViz: Pizza
Show class
Show children
Show parents
Show all classes
Hide class
Hide children
Hide classes
Asserted model
Inferred model
Hot
Medium
Mild
IceCream
PizzaBase
PizzaTopping
Pizza
is-a
CheeseyPizza
ThinAndCrispyPizza
NamedPizza
SpicyPizzaEquivalent
VegetarianPizzaEquivalent1
VegetarianPizza
NonVegetarianPizza
InterestingPizza
QuattroFormaggi
Napoletana
Rosa
PolloAdAstra
Soho
Margherita
Fiorentina
LaReine

Baze znanja

- Cyc
  - Največji sistem za delo z znanjem
  - Temelji na PR
  - Sklepanje s pravili
- Predstavitev znanja
  - Konceptualne mreže (ontologije)
  - Višja ontologija, osnovne teorije, specifične teorije
  - Predefinirane semantične relacije

Baza znanja Cyc



Objektno-relacijski model (1)

- Relacijski model
  - Codd, 1970
  - Matematične osnove: n-terice in relacije
  - Izjavni račun => relacijski račun
  - Algebra relacij
  - Deklarativni povpraševalni jezik SQL2
    - De-facto standard
    - Relacijska kompletnost jezika

Objektno-relacijski model (2)

- Relacijski model
  - Proceduralni programski jezik SQL2
  - Shranjene procedure
  - Računanje s tabelami
  - Računsko kompleten jezik
  - Vsebuje pravila
    - Prožilci
    - Realizacija AI ekspertnih sistemov
    - Še precej problemov ...

Objektno-relacijski model (3)

- SQL3
  - Objektno-relacijski model
  - Abstraktni Podatkovni Tipi = Razredi
    - Uporabniško definirani tipi
    - Metode, dedovanje, polimorfizem,
  - Kompleksni atributi
    - n-terice, množice, polja, tabele, itd.
    - Bogat nabor operacij

Objektno-relacijski model (4)

- SQL3
  - Uporabljeni programski modeli
    - Proceduralni gradniki
    - Objektni gradniki
    - Pravila in prožilci
    - Rekurzija, fiksne točke
  - Novi tipi
    - Multimedia: blob, clob
    - Intervali
    - Drevesa, grafi

Logični model (1)

- Logika uporabljena za
  - predstavitev sheme in podatkov
  - povpraševalni jezik
- Največ modelov sloni na predikatnem računu
- Podmnožica predikatnega računa
  - Hornovi stavki
  - Prolog

Logični model (2)

- LDM (Kuper,Vardi)
  - Semantični model
- Datalog (Ullman)
  - Prolog nad podatkovnimi bazami
- F-Logic (Kifer)
  - Objektni model
  - Prolog

Literatura

- I. Savnik, Prosojnice za predmet “Podatkovni modeli in jeziki”, Spletna učilnica, 2013/14.
- Literatura za posamezno poglavje bo na voljo v spletni učilnici.