UNIVERSITY OF ABERDEEN

Knowledge-Based Systems

Summary (Ontology Part)

Jeff Z. Pan

http://homepages.abdn.ac.uk/jeff.z.pan/pages/

Lecture Outline

- Knowledge Based Systems
  - What and Why
  - Applications and Further Challenges
  - Lifecycle of KBS
- **The TrOWL Award Announcement**
- Exam

Knowledge-Based Systems

- A knowledge-based system (KBS) is:
  - an program;
  - capable of making use of knowledge about certain domain(s) to make inference;
  - with a view to solving problems and giving advice.

- Sometimes knowledge bases also known as Semantic databases

RDB -> RDF

- An RDF statement is a data unit with global and linkable ids for data and schema

Student ID	Name	take-course
p001	John	cs3019
p002	Tom	cs3023

  - [csd:p001 rdf:type csd:Student .]
  - [csd:p002 rdf:type csd:Student .]
  - [csd:p001 csd:name “John” .]
  - [csd:p002 csd:name “Tom” .]
  - [csd:p001 csd:take-course csd:cs3019 .]
  - [csd:p002 csd:take-course csd:cs3023 .]

Schema in a Database System

- A database system includes some schema constraints, such as the foreign key constraint

Student ID	Name	take-course
p001	John	~~cs3015~~
p002	Tom	cs3025

Course ID	Title	coordinator
cs3017	AIS	AS
cs3025	KBS	JP

Schema in a Knowledge Based System

1) Allow schema constraints, such as DisjointClasses (UndgStudent MastStudent)

UndgStudent ID	Name	take-course
csd:p001	John	csd:cs3014
~~csd:p002~~	Tom	csd:cs3025

MastStudent ID	Name	take-course
csd:p008	Yuan	csd:cs5010
~~csd:p002~~	Tom	csd:cs5017

Schema in a Knowledge Based System

2) Allow some reasoning based on axioms (open world assumption), such as SubClassOf (MastStudent Student)

Student ID	Name	take-course
csd:p001	John	csd:cs3015
csd:p002	Tom	csd:cs3025

MastStudent ID	Name	take-course
csd:p008	Yuan	csd:cs5010
csd:p002	Tom	csd:cs5017

thus all the students include csd:p001, csd:p002, and csd:p008

rdf:range and Foreign Key: Revisit

- They are quite similar but not exactly the same,
  - due to the difference between open and closed world assumptions

Student ID	Name	take-course
p001	John	~~cs3015~~
p002	Tom	cs3025

Course ID	Title	coordinator
cs3017	AIS	AS
cs3025	KBS	JP

- Semantics of rdfs:range ([rdfs3])
  - [p rdfs:range D .] [a p b .] => [b rdf:type D .]

How about Primary Key

- RDF is NOT expressive enough to represent primary key
- We need OWL (Web Ontology Language) for that

- Example with Friend Of a Friend (FoaF) ontology
  – foaf:OnlineAccount owl:hasKey

    (foaf:accountName

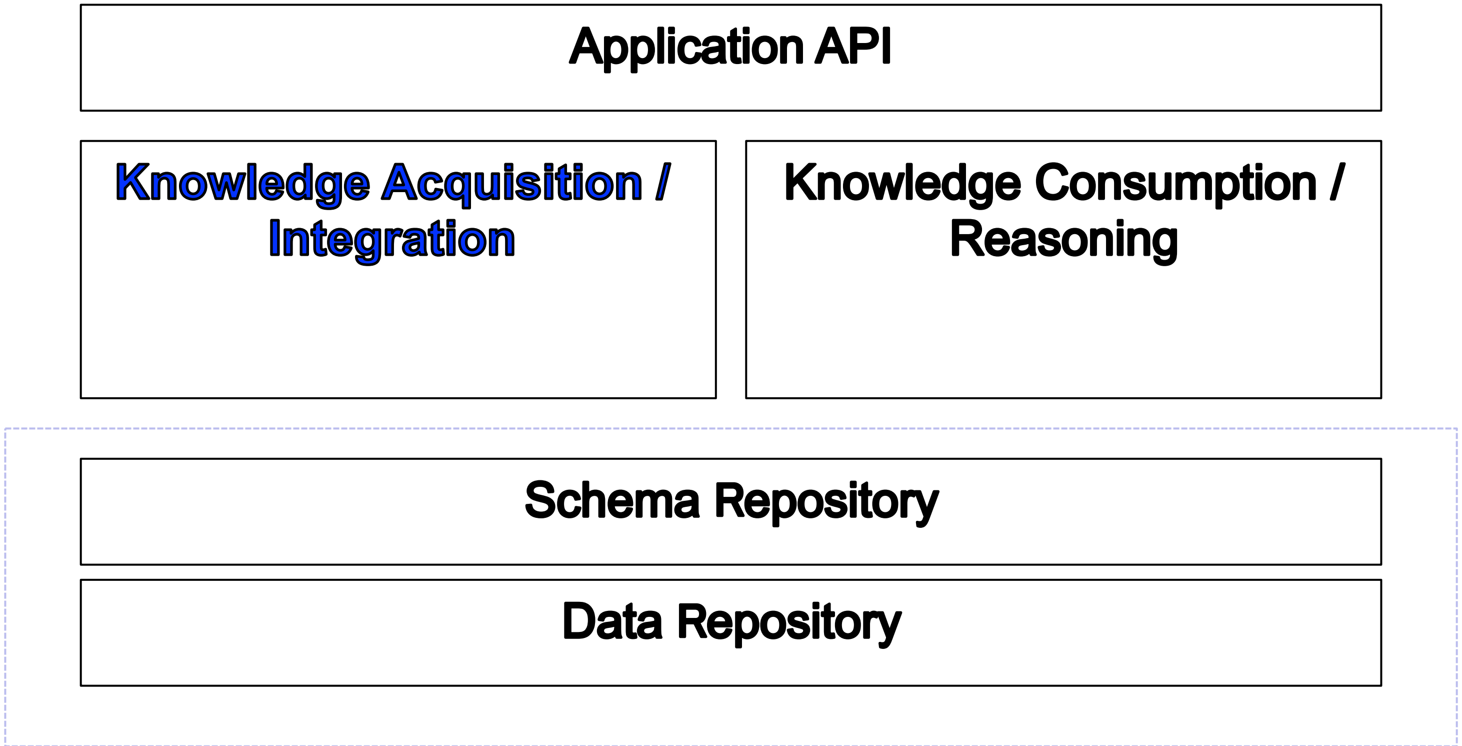
    foaf:accountServiceHomepage)

Why Knowledge Bases

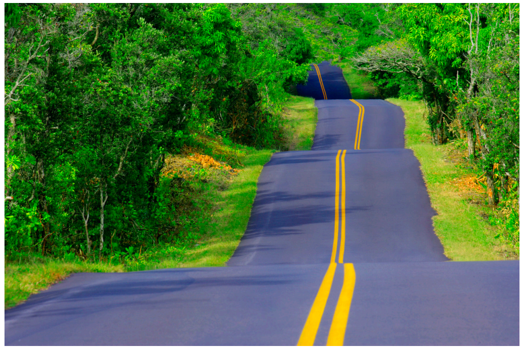
- Using an open world assumption
- There is a semantic gap between data and queries
- Knowledge bases used to store knowledge and experience
  - Many well-known ontology
    - E.g. SNOMED CT
  - Semantic applications populate terminologies with data
    - E.g. Chintan Patel et al (SWJ2007) populated SMOMED CT with 59 million ABox assertions.

Scalable Solution: Olympic 2012 web site

Architecture of Knowledge Based Systems



Course Roadmap

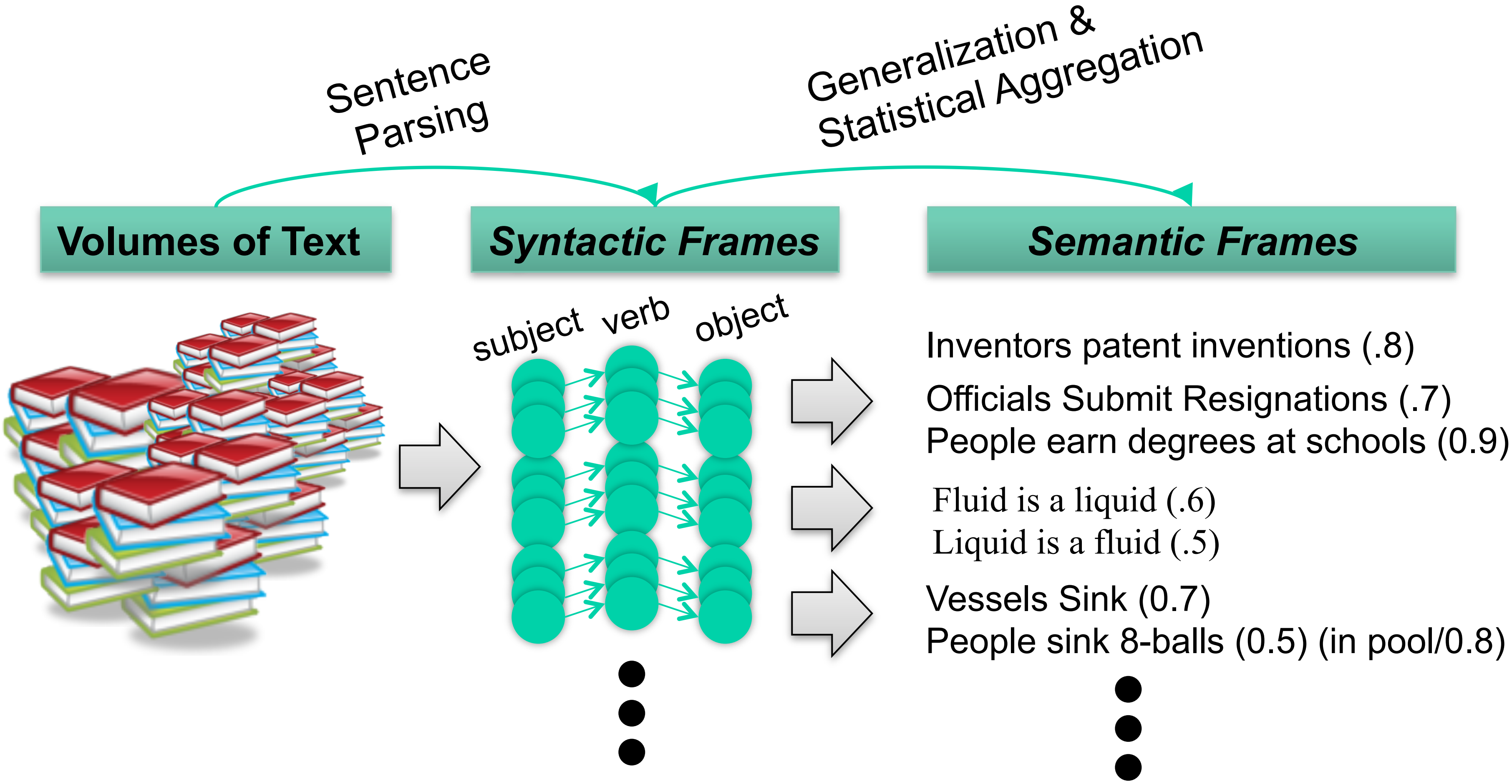


- Foundation
  - KR, ontology and rule; set theory
- Knowledge capture
- Knowledge representation
  - Ontology: Semantic Web standards RDF and OWL, Description Logics
  - Rule: Jess
- Knowledge reasoning
  - Ontology: formal semantics, tableaux algorithm
  - Rule: forward chaining, backward chaining
- Knowledge reuse and evaluation
- Meeting the real world
  - Jess and Java, Jess Efficiency, Invited talk

Interview: Knowledge Capture

- Unstructured interview
  - No detailed agenda
  - Few constraints
  - Delivers diverse, incomplete data
- Structured interview
  - Knowledge engineer plans and directs the session
    - Takes form of provider-elicitor dialogue
  - Delivers more focused expertise data

Automatic Learning for Watson



[Slide Credit: Joel Farrell]

Microsoft Research

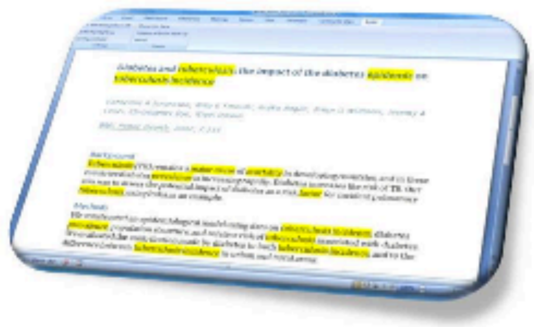
Search Microsoft Research

Home | Our Research | Connections | Careers

Worldwide Labs | Research Areas | Research Groups

> Projects > Ontology Add-in for Word

Ontology Add-in for Word



Microsoft Research Connections' goal with this project is to enable communities who maintain ontologies to more easily experiment and to enhance the experience of authors who use Microsoft Word for content creation, incorporating semantic knowledge into the content.

This add-in should simplify the development and validation of ontologies, by making ontologies more accessible to a wide audience of authors and by enabling semantic content to be integrated in the authoring experience, capturing the author's intent and knowledge at the source, and facilitating downstream discoverability.

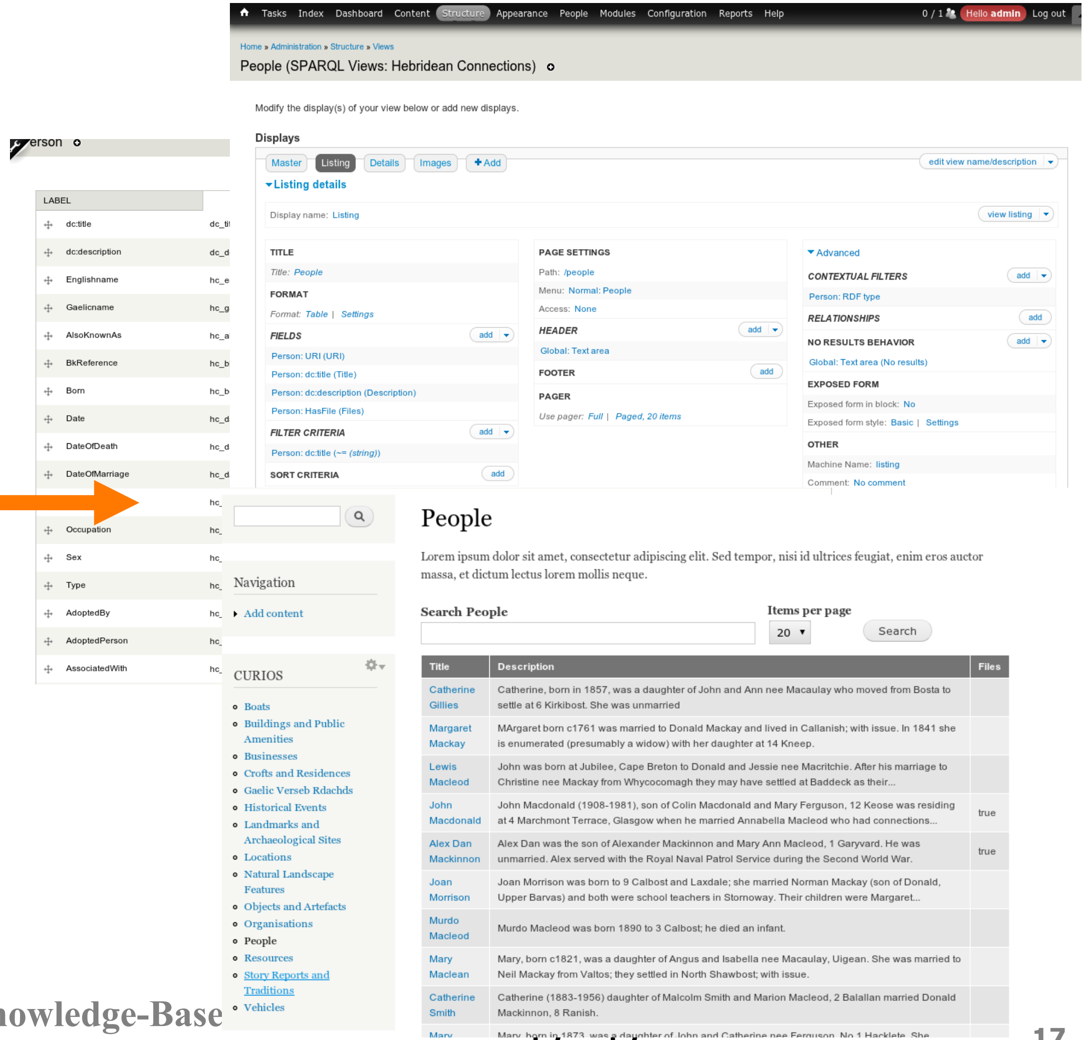
The goal of the add-in is to assist scientists in writing a manuscript that is easily integrated with existing and pending electronic resources. The major aims of this project are to add semantic information as XML mark-up to the manuscript using

![UNIVERSITY OF ABERDEEN]

CURIOS: OWL Adaptor for CMS

Domain ontology

LDCMS configuration (in php)



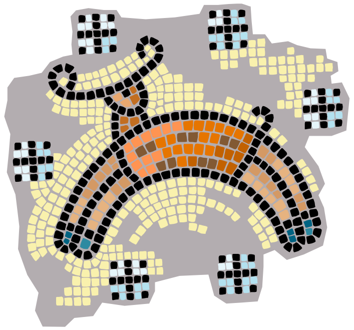
Competency Question

- A typical CQ: Which pizza has some cheese topping?
- When building an ontology, it is less important what the answer is
  - The ontology can be extended and modified so the answer is not final
- More importantly is whether the CQ can be answered meaningfully
  - The ability to ***answer CQs meaningfully*** can be regarded as a functional requirement of the ontology

Knowledge Representation: RDF and OWL

- RDF (Resource Description Framework)
  - Modern version of semantic network
  - Basic building block: Subject-property-value triple
  - Modern version of Semantic Network
  - Syntax (**exam only requires the use of N3 syntax**)
    - XML
    - Notation 3 (N3) syntax: [Jeff teach CS3025 .]
- OWL (Web Ontology Language)
  - More expressive power, such as
    - Disjointness of classes
      - DisjointClasses (csd:Male csd:Female)
    - Equivalent individuals
      - SameIndividual (csd:jpan w3:jpan)
    - Boolean combination of classes
      - Class (Person complete unionOf (csd:Male csd:Female))
  - Well known techniques for transforming semantic networks to ontologies
  - Based on Description Logics (**DL tables will be provided in the exam**)

DL Exercise



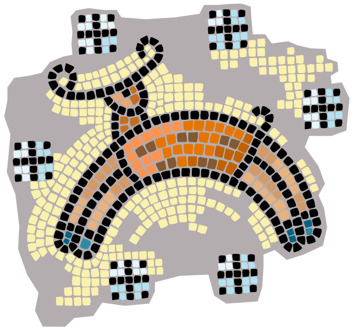
Q: Write down the following OWL axioms in DL syntax

1. ObjectProperty (eats domain (Animal)
   range (LivingThing))

2. ObjectProperty (owns domain (Person)
   range (intersectionOf (LivingThing
   complementOf (Person))))

Interpretations of Axioms

DL Exercise



Q: Write down the following OWL axioms in DL syntax

ObjectProperty(R super(R_1) ... super(R_n)	$R \sqsubseteq R_i$
domain(C_1) ... domain(C_k)	$\geqslant 1 R \sqsubseteq C_i$
range(C_1) ... range(C_h))	$\top \sqsubseteq \forall R.C_i$

1. ObjectProperty (eats domain (Animal) range (LivingThing))

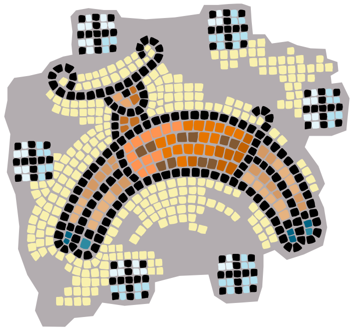
$$>=1 \text{ eat} \sqsubseteq \text{Animal}$$

$$T \sqsubseteq \forall \text{eat.LivingThing}$$

Interpretations of Descriptions

Abstract Syntax	DL Syntax	Semantics
Class(A)	A	$\mathsf{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Class(owl:Thing)	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
Class(owl:Nothing)	$\bot$	$\bot^{\mathcal{I}} = \emptyset$
intersectionOf($C_1, C_2, \ldots$)	$C_1 \sqcap C_2$	$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
unionOf($C_1, C_2, \ldots$)	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
complementOf(C)	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
oneOf($o_1, o_2, \ldots$)	$\{o_1\} \sqcup \{o_2\}$	$(\{o_1\} \sqcup \{o_2\})^{\mathcal{I}} = \{o_1{}^{\mathcal{I}}, o_2{}^{\mathcal{I}}\}$
restriction(R someValuesFrom(C))	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x,y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
restriction(R allValuesFrom(C))	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x,y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
restriction(R hasValue(o))	$\exists R.\{o\}$	$(\exists R.\{o\})^{\mathcal{I}} = \{x \mid \langle x, \mathsf{o}^{\mathcal{I}}\rangle \in R^{\mathcal{I}}\}$
restriction(R minCardinality(m))	$\geqslant mR$	$(\geqslant mR)^{\mathcal{I}} = \{x \mid \sharp\{y.\langle x,y\rangle \in R^{\mathcal{I}}\} \geq m\}$
restriction(R maxCardinality(m))	$\leqslant mR$	$(\leqslant mR)^{\mathcal{I}} = \{x \mid \sharp\{y.\langle x,y\rangle \in R^{\mathcal{I}}\} \leq m\}$
restriction(T someValuesFrom(u))	$\exists T.u$	$(\exists T.u)^{\mathcal{I}} = \{x \mid \exists t.\langle x,t\rangle \in T^{\mathcal{I}} \wedge t \in u^{\mathbf{D}}\}$
restriction(T allValuesFrom(u))	$\forall T.u$	$(\forall T.u)^{\mathcal{I}} = \{x \mid \exists t.\langle x,t\rangle \in T^{\mathcal{I}} \rightarrow t \in u^{\mathbf{D}}\}$
restriction(T hasValue(w))	$\exists T.\{w\}$	$(\exists T.\{w\})^{\mathcal{I}} = \{x \mid \langle x, w^{\mathbf{D}}\rangle \in T^{\mathcal{I}}\}$
restriction(T minCardinality(m))	$\geqslant mT$	$(\geqslant mT)^{\mathcal{I}} = \{x \mid \sharp\{t \mid \langle x,t\rangle \in T^{\mathcal{I}}\} \geq m\}$
restriction(T maxCardinality(m))	$\leqslant mT$	$(\leqslant mT)^{\mathcal{I}} = \{x \mid \sharp\{t \mid \langle x,t\rangle \in T^{\mathcal{I}}\} \leq m\}$
ObjectProperty(S)	S	$S^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
ObjectProperty(S' inverseOf(S))	S^{-}	$(S^{-})^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
DatatypeProperty(T)	T	$T^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}$

DL Exercise



Q: Write down the following OWL axioms in DL syntax

ObjectProperty(R super(R_1) ... super(R_n) domain(C_1) ... domain(C_k) range(C_1) ... range(C_h))	$R \sqsubseteq R_i$ $\geqslant 1 R \sqsubseteq C_i$ $\top \sqsubseteq \forall R.C_i$

intersectionOf($C_1, C_2, \ldots$)	$C_1 \sqcap C_2$
unionOf($C_1, C_2, \ldots$)	$C_1 \sqcup C_2$
complementOf(C)	$\neg C$

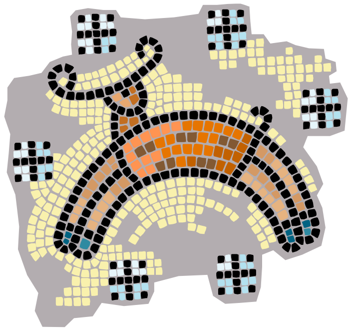
2. ObjectProperty (owns domain (Person) range (intersectionOf (LivingThing complementOf (Person))))

$>=1 own \sqsubseteq Person$

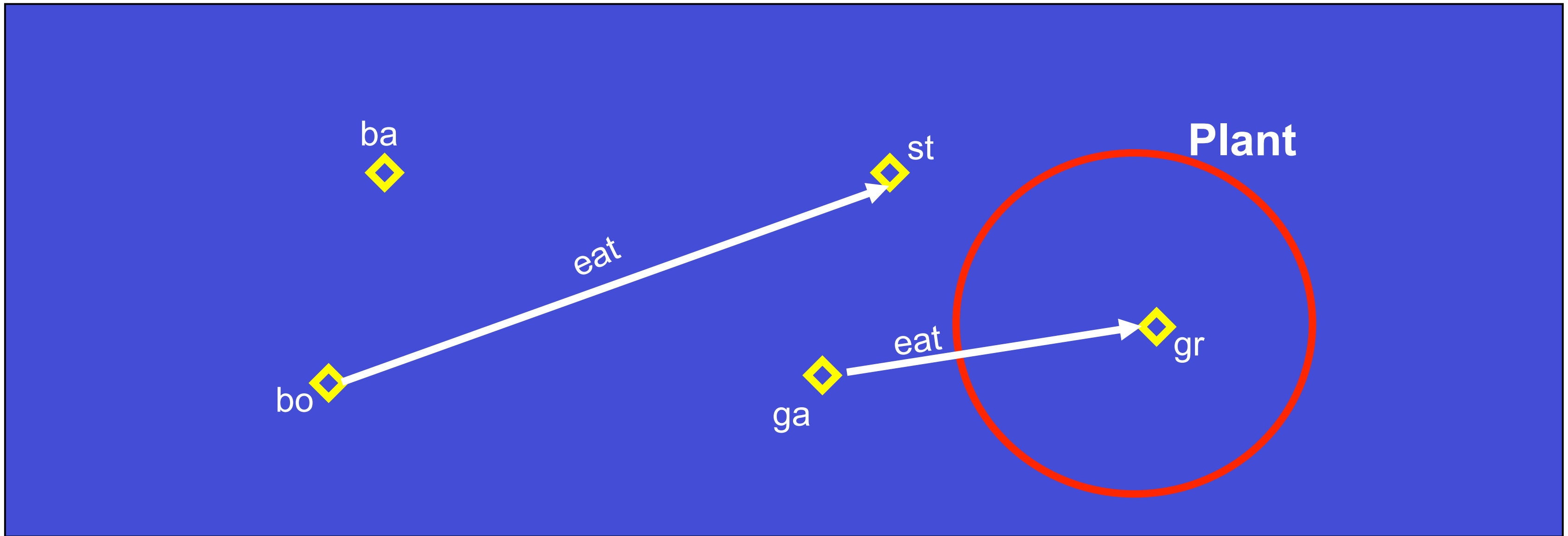
$T \sqsubseteq \forall eat.(LivingThing \sqcap \neg Person)$

Understanding DL Semantics

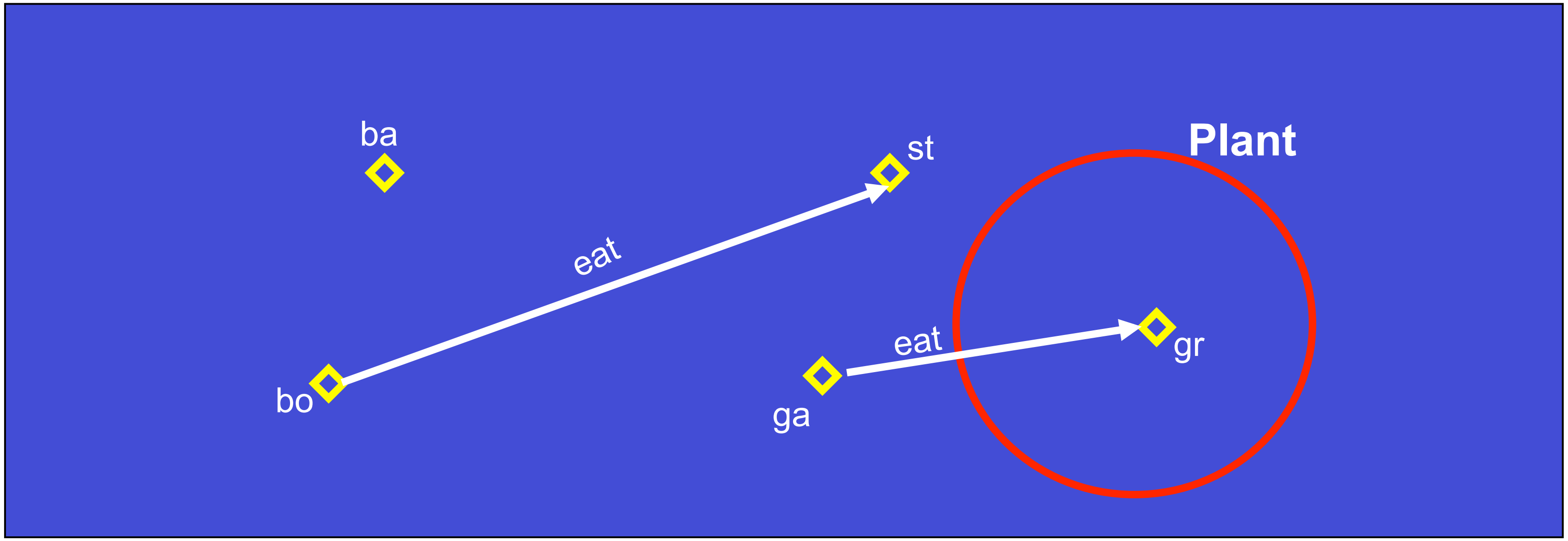
Q: Given the following interpretation



1. Δ^I ={ga,bo,ba,gr,st}
2. Plant^I= {gr}
3. eat^I=$\{\langle ga,gr \rangle,\langle bo,st \rangle\}$
4. partOf^I= ;



DL Semantics Exercise



5. $(\exists partOf.Plant)^{I}$

$= \{x|\exists y.\langle x,y\rangle \in partOf^{I} \wedge y \in Plant^{I}\}$ (Semantics of $\exists R.C$)

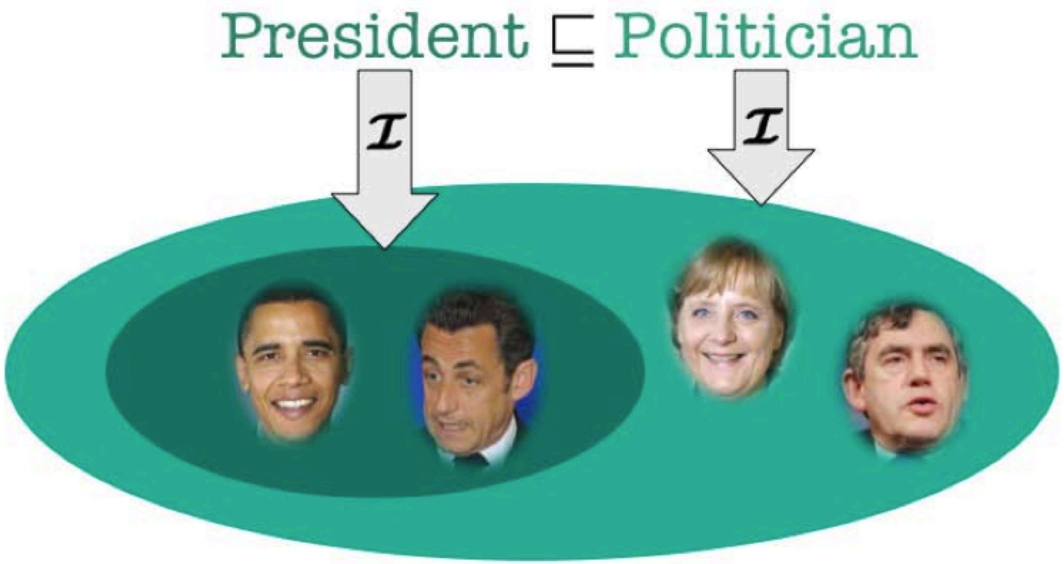
$= \{x|\exists y.\langle x,y\rangle \in \emptyset; \wedge y \in \{gr\}\}$ (from 2,4)

$= \emptyset$

"Every individual of the class restriction must have at least one partOf relation with an instance of the class **Plant**"

Axioms

- Axioms are used to “filter out” invalid interpretations from valid ones
  - An interpretation I is a model for an ontology O if it satisfies all its axioms
  - An ontology O is consistent if it has some model (valid interpretation).



DL Reasoning Services

- Basic OWL DL Reasoning services
  - Class subsumption checking
    - subsume matching, plugin matching
  - Class equivalence checking
    - exact matching
  - Class satisfiability checking
    - intersection matching
  - Ontology consistency checking
  - Class instance checking
  - Property instance checking

UNIVERSITY OF ABERDEEN

Prove it!

Prove if the following equivalences hold

$$C \sqcap D \equiv D \sqcap C$$

$(C \sqcap D)^I = C^I \cap D^I$

$(D \sqcap C)^I = D^I \cap C^I = C^I \cap D^I$

$$(C \sqcap D) \sqcap E \equiv C \sqcap (D \sqcap E)$$

$$C \sqcap (D \sqcup E) \equiv (C \sqcap D) \sqcup E$$

Class Equivalence

Two class descriptions are called "equivalent" (written as C ≡ D) if for every single interpretation I we have $C^I = D^I$.

$$
\begin{array}{cc}
C \sqcap D \equiv D \sqcap C & C \sqcup D \equiv D \sqcup C \\
(C \sqcap D) \sqcap E \equiv C \sqcap (D \sqcap E) & (C \sqcup D) \sqcup E \equiv C \sqcup (D \sqcup E) \\
C \sqcap C \equiv C & C \sqcup C \equiv C
\end{array}
$$

$$
\begin{array}{cc}
(C \sqcup D) \sqcap E \equiv (C \sqcap E) \sqcup (D \sqcap E) & (C \sqcup D) \sqcap C \equiv C \\
(C \sqcap D) \sqcup E \equiv (C \sqcup E) \sqcap (D \sqcup E) & (C \sqcap D) \sqcup C \equiv C
\end{array}
$$

$$
\begin{array}{ccc}
 & \neg\exists r.C \equiv \forall r.\neg C & \geqslant 0 r.C \equiv \top \\
\neg\neg C \equiv C & \neg\forall r.C \equiv \exists r.\neg C & \geqslant 1 r.C \equiv \exists r.C \\
\neg(C \sqcap D) \equiv \neg D \sqcup \neg C & \neg\leqslant n r.C \equiv \geqslant (n+1) r.C & \leqslant 0 r.C \equiv \forall r.\neg C \\
\neg(C \sqcup D) \equiv \neg D \sqcap \neg C & \neg\geqslant (n+1) r.C \equiv \leqslant n r.C &
\end{array}
$$

UNIVERSITY OF ABERDEEN

Class Subsumption Checking

Class(A partial C_1 ... C_n)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$
Class(A complete C_1 ... C_n)	$A \equiv C_1 \sqcap \ldots \sqcap C_n$

- Question: Given the following ontology O,
  - Class (C complete complementOf (
    restriction (eats someValuesFrom (Plant))))
  - Class (D complete restriction (eats
    allValuesFrom (complementOf (Plant)))

  Does O entail Class(C partial D)?

$$\neg\exists r.C \equiv \forall r.\neg C$$
$$\neg\forall r.C \equiv \exists r.\neg C$$

Does O entails $\neg\exists eats.Plant \sqsubseteq \forall eats.(\neg Plant)$? **true**

Tableaux Algorithm

- The first sound and complete algorithm for expressive DLs
  - Ontology Consistency Checking
  - Class Satisfiability Checking
  - Instance Checking
  - Subsumption Checking
- Basic idea: Build an interpretation
  - A **tableau** is a representative of an interpretation
  - We can construct an interpretation based on a tableau

Key Steps

1. Initialise the tableau
   – depending on the reasoning tasks
2. Repair the initial tableau by applying expansion rules
   – so as to add new information into the tableau
   – this might require backtracking
3. If the tableau satisfy all the axioms, returns Consistent
4. If every possible attempt repair results in some contradiction, returns Inconsistent
   – Contradiction: $\{A, \neg A\} \subseteq L(x)$ or $\{\perp\} \subseteq L(x)$ ($\perp$ is bottom, interpreted as **empty set**)

Class Instance Checking

- Given an ontology O, a class C and an individual x, check if for every interpretation **I** of O, x^I is in C^I

- Question: given the following ontology O,
  - Class (OldLady partial

    restriction (hasPet allValuesFrom (Cat)))
  - Individual (Minnie type (OldLady)

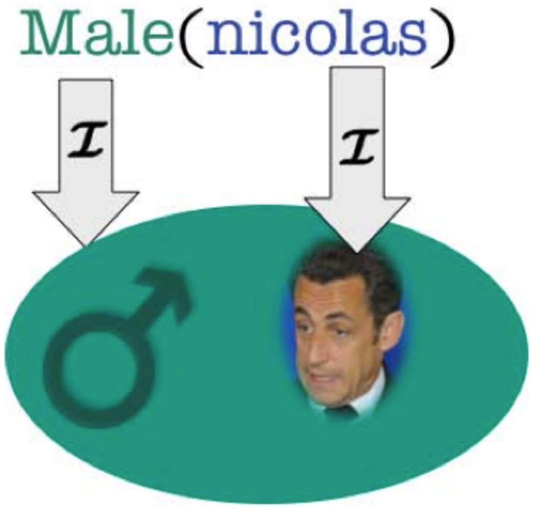
    value (hasPet Tom))
- Does O entail Individual (Tom type (Cat)) ?

Class Instance Checking

- Can we reduce class instance checking to another reasoning task?
- How about Ontology Consistency Checking
  - If O entails C(x), then in every interpretation I of O, we have x^I is in C^I
  - It means $O \cup O\{\neg C(x)\}$ is inconsistent

Example

- If an ontology O entails Male (nicolas)
  - then in every interpretation I of O
  - we have $nicolas^I \in Male^I$
- Now if we extend O to O’ with a new axiom
  - ¬Male(nicolas) (*)
- How to construct an interpretation I’ for O’?
  - as all interpretations of O’ should satisfy O
  - we could start from interpretations of O
- It is easy to see that I’ does not exist
  - If I’ does not satisfy O, then it does not satisfy O’ either
  - If I’ satisfies O, then it does not satisfy ¬Male(nicolas)



Expansion Rules for Some Constructors

$x \bullet \{C_1 \sqcap C_2, \ldots\}$	$\rightarrow_{\sqcap}$	$x \bullet \{C_1 \sqcap C_2, C_1, C_2, \ldots\}$
$x \bullet \{C_1 \sqcup C_2, \ldots\}$	$\rightarrow_{\sqcup}$	$x \bullet \{C_1 \sqcup C_2, C, \ldots\}$ for $C \in \{C_1, C_2\}$
$x \bullet \{\exists R.C, \ldots\}$	$\rightarrow_{\exists}$	$x \bullet \{\exists R.C, \ldots\}$ $R \downarrow$ $y \bullet \{C\}$
$x \bullet \{\forall R.C, \ldots\}$ $R \downarrow$ $y \bullet \{\ldots\}$	$\rightarrow_{\forall}$	$x \bullet \{\forall R.C, \ldots\}$ $R \downarrow$ $y \bullet \{C, \ldots\}$

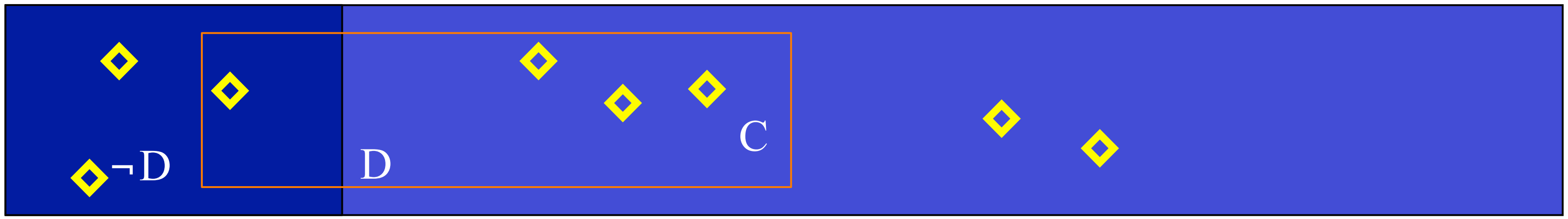
Class Instance Checking

- Question: given the following ontology O,
  - OldLady $\sqsubseteq$ $\forall$hasPet.Cat
  - OldLady(Minnie)
  - hasPet(Minnie, Tom)

  ¬Cat(Tom)
- Does O entail Individual (Tom type (Cat)) ?

From Subsumption Checking to Class Unsatisfiability Checking

- Some reasoning services can be reduced to each other
- E.g. class subsumption checking to class (un)satisfiability checking
  - $O \models C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable ?



Example: Class SubsumptionChecking

- Question: Given the following ontology O,
  - Class (Chinese partial Person)
  - Class (English partial Person)
  - Class (Confucian partial Chinese)
  - Class (Confucian partial English)
- To check O |= Confucian $\sqsubseteq$ Person, we need to check if (Confucian $\sqcap \neg$Person) is unsatisfiable

Assessments

- 25% Continuous Assessment
  - Demonstrate practical skills related to ontology and rule, including building a **reasoner**
  - **Challenge: implementation of the inverse object property expansion rule.**
- **The TrOWL Award**
  - **for the best reasoner among the assessment submissions**

The TrOWL Reasoner

- http://www.slideshare.net/jeffpan_sw/the-rise-of-approximate-ontology-reasoning-is-it-mainstream-yet

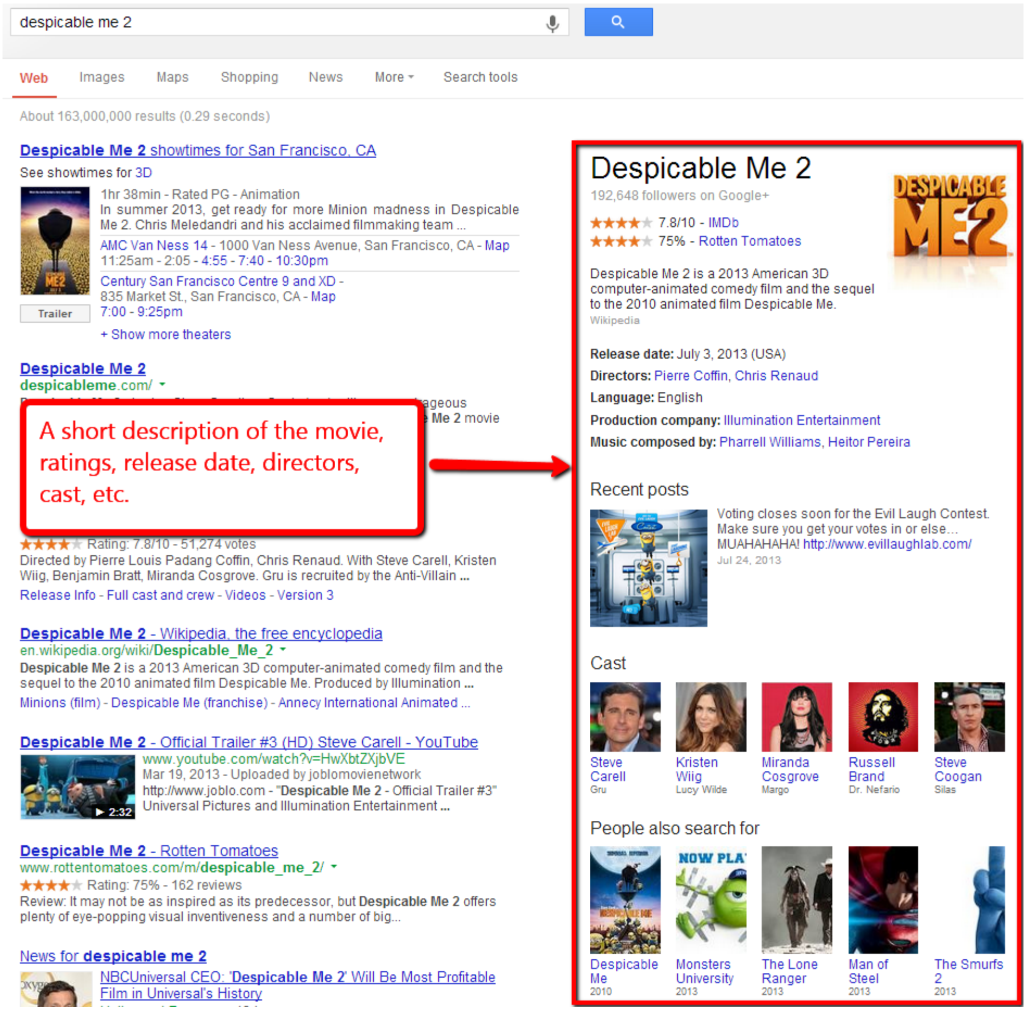
Winner(s) of the TrOWL Award

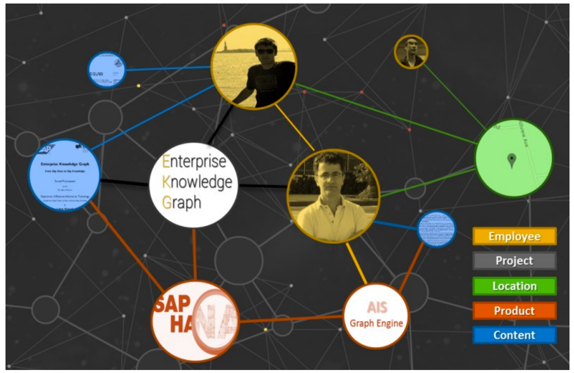
- Winners are ...

Exam

- 75% of overall marking
- Apply CS3025 learning outcomes
  – 2 hours
  – 2 questions: one on ontology and the other on Jess

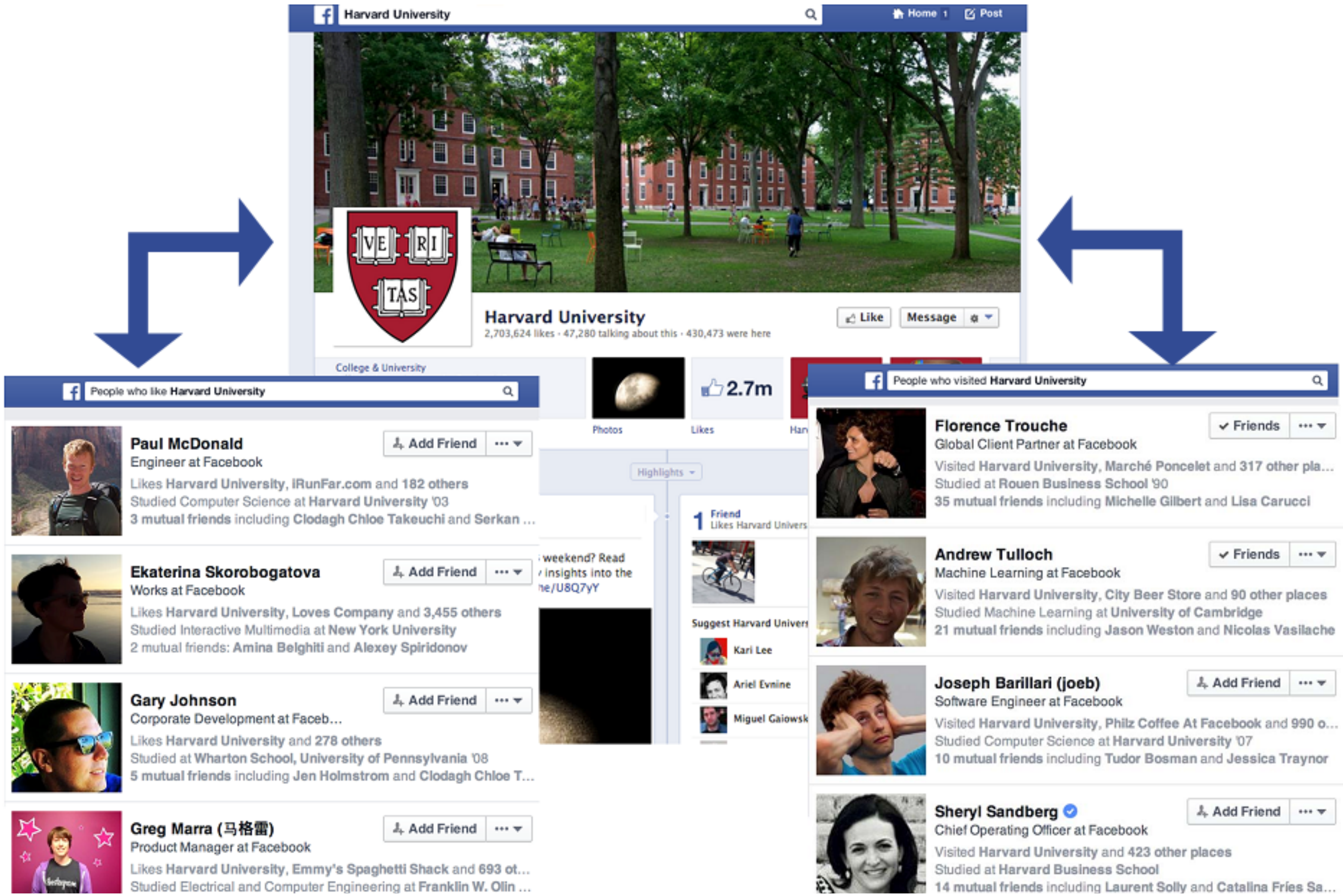
CS4046 Semantic Web Engineering





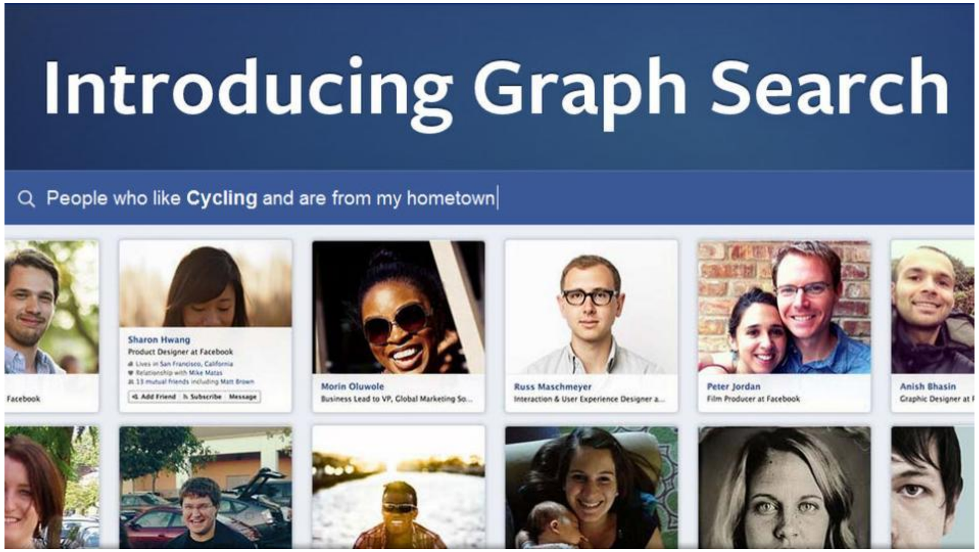
[Credit: H Wang]

CS4046 Semantic Web Engineering



CS4046 Semantic Web Engineering

Structured search within the graph



CS4046 Semantic Web Engineering

A. What is the computer system that played against human opponents on "Jeopardy"… and won.

Why Jeopardy?

The game of *Jeopardy!* makes great demands on its players – from the range of topical knowledge covered to the nuances in language employed in the clues. The question IBM had for itself was "is it possible to build a computer system that could process big data and come up with sensible answers in seconds—so well that it could compete with human opponents?"

[Slide Credit: Dalal et al.]

Where to go from here

- Reading
  - slides
- Understanding
  - practicals
  - sample exam questions
- Q&A
  - email me your question
  - I will send my answer to the mailing list

Please feel free to contact if you would like to do an Honours project related to KBS