CPE/CSC 481: Knowledge-Based Systems

Franz J. Kurfess

Computer Science Department
California Polytechnic State University
San Luis Obispo, CA, U.S.A.

Usage of the Slides

- **these slides are intended for the students of my CPE/CSC 481 “Knowledge-Based Systems” class at Cal Poly SLO**
  - if you want to use them outside of my class, please let me know (fkurfess@calpoly.edu)
- **I usually put together a subset for each quarter as a “Custom Show”**
  - to view these, go to “Slide Show => Custom Shows”, select the respective quarter, and click on “Show”
- **To print them, I suggest to use the “Handout” option**
  - 4, 6, or 9 per page works fine
  - Black & White should be fine; there are few diagrams where color is important

Course Overview

- **Introduction**
  - Knowledge-Based Systems (KBS), Expert Systems (ES)
  - Data/Information/Knowledge
- **Knowledge Representation**
  - Semantic Nets, Rules, Frames, Scripts, Logic, RDF
- **Reasoning and Inference**
  - Predicate Logic, Description Logics, Inference Methods, Resolution
- **Reasoning with Uncertainty**
  - Probability, Bayesian Decision Making
- **Approximate Reasoning**
  - Fuzzy Logic
- **Knowledge Exchange**
  - Capture, Transfer, Distribution
- **Knowledge Retrieval**
  - Search, Queries, Data Mining
- **KBS Implementation**
  - Unification, Pattern Matching, Salience, Rete Algorithm
- **KBS Examples**
  - CLIPS/Jess, Prolog, Semantic Web Technologies
- **Conclusions and Outlook**

Overview
Knowledge Representation

- **Motivation**
- **Objectives**
- **Chapter Introduction**
  - Review of relevant concepts
  - Overview new topics
  - Terminology
- **Knowledge and its Meaning**
  - Epistemology
  - Types of Knowledge
  - Knowledge Pyramid
- **Knowledge Representation Methods**
  - Production Rules
  - Semantic Nets
  - Schemata and Frames
  - Logic
  - Semantic Web and KR
- **Ontologies**
  - OWL
  - RDF
- **Important Concepts and Terms**
- **Chapter Summary**

Logistics

- Term Project
- Lab and Homework Assignments
- Quizzes or Term Paper?
- Grading

Motivation

- KBS are useless without the ability to represent knowledge
- different knowledge representation schemes may be appropriate
  - depending on tasks and circumstances
- knowledge representation schemes and reasoning methods must be coordinated

Objectives

- **know the basic principles and concepts for knowledge representation**
  - knowledge · information · data
  - meaning
- **be familiar with the most frequently used knowledge representation methods**
  - logic, rules, semantic nets, schemata
    - differences between methods, advantages, disadvantages, performance, typical scenarios
- **understand the relationship between knowledge representation and reasoning**
  - syntax, semantics
  - derivation, entailment
- **apply knowledge representation methods**
  - usage of the methods for simple problems

Knowledge and its Meaning

Epistemology
Types of Knowledge
Knowledge Pyramid

Epistemology

- **the science of knowledge**
- **EPISTEMOLOGY (Gr. episteme, "knowledge"; logos, "theory"),**
- **branch of philosophy concerned with the theory of knowledge.**
  - The main problems with which epistemology is concerned are the definition of knowledge and related concepts, the sources and criteria of knowledge, the kinds of knowledge possible and the degree to which each is certain, and the exact relation between the one who knows and the object known.

[Infopedia 1996]

Knowledge Definitions

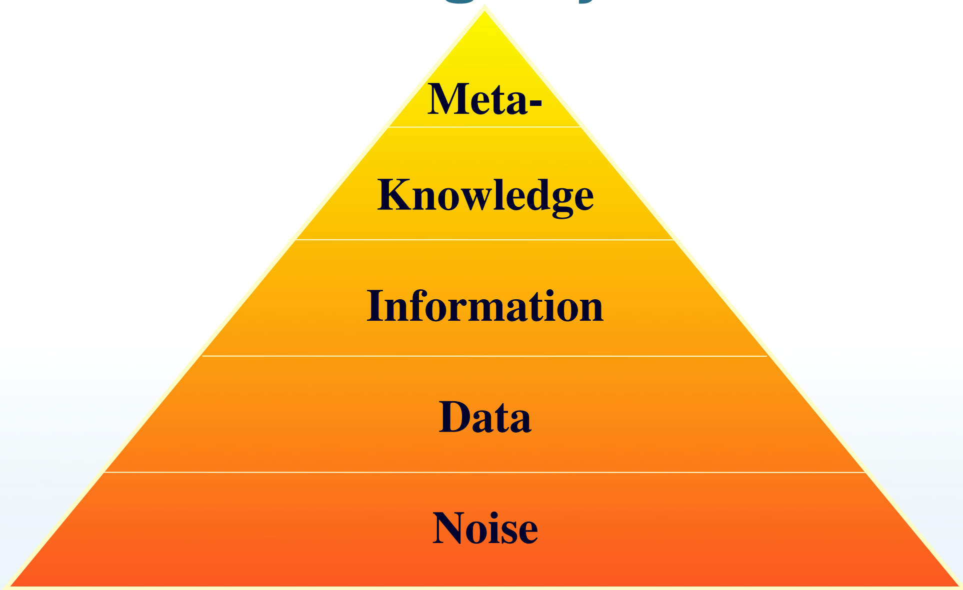
- knowlaedge \'nS-lij\ n [ME knowlege, fr. knowlechen to acknowledge, irreg. fr. knowen] (14c)
- 1 obs : cognizance
- 2 a
- (1) : the fact or condition of knowing something with familiarity gained through experience or association
- (2) : acquaintance with or understanding of a science, art, or technique
- b
- (1) : the fact or condition of being aware of something
- (2) : the range of one's information or understanding <answered to the best of my 4>
- c : the circumstance or condition of apprehending truth or fact through reasoning : cognition
- d : the fact or condition of having information or of being learned <a man of unusual 4>
- 3 archaic : sexual intercourse
- 4 a : the sum of what is known : the body of truth, information, and principles acquired by mankind
- b archaic : a branch of learning syn knowledge, learning, erudition, scholarship mean what is or can be known by an individual or by mankind. knowledge applies to facts or ideas acquired by study, investigation, observation, or experience <rich in the knowledge of human nature>. learning applies to knowledge acquired esp. through formal, often advanced, schooling <a book that demonstrates vast learning >. erudition strongly implies the acquiring of profound, recondite, or bookish learning <an erudition unusual even in a scholar>. scholarship implies the possession of learning characteristic of the advanced scholar in a specialized field of study or investigation <a work of first-rate literary scholarship >.

[Merriam-Webster, 1994]

Types of Knowledge

- **a priori knowledge**
  - comes before knowledge perceived through senses
  - considered to be universally true
- **a posteriori knowledge**
  - knowledge verifiable through the senses
  - may not always be reliable
- **procedural knowledge**
  - knowing how to do something
- **declarative knowledge**
  - knowing that something is true or false
- **tacit knowledge**
  - knowledge not easily expressed by language

Knowledge Pyramid



Knowledge Representation Methods

Production Rules
Semantic Nets
Schemata and Frames
Logic

CAL POLY

Production Rules

- **frequently used to formulate the knowledge in expert systems**
- **a formal variation is Backus-Naur form (BNF)**
  - metalanguage for the definition of language syntax
  - a grammar is a complete, unambiguous set of production rules for a specific language
  - a parse tree is a graphic representation of a sentence in that language
  - provides only a syntactic description of the language
    - not all sentences make sense

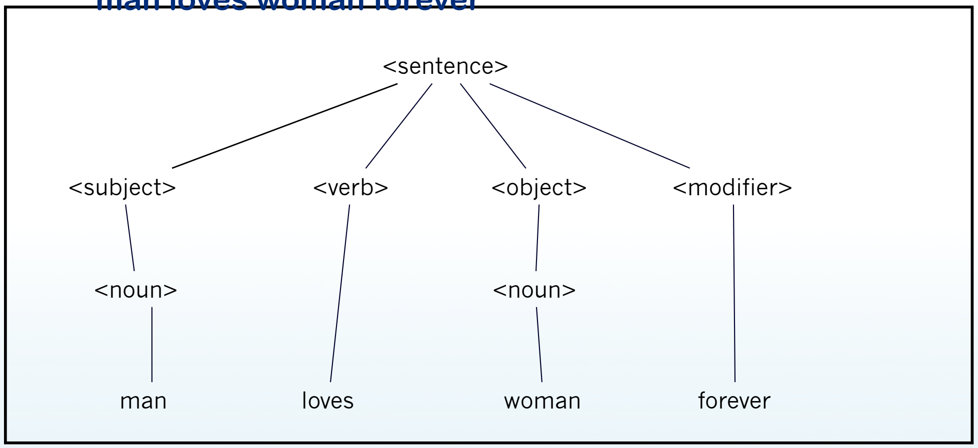
Example 1 Production Rules

- **for a subset of the English language**

```
<sentence>    -> <subject> <verb> <object> <modifier>
<subject>     -> <noun>
<object>      -> <noun>
<noun>        -> man | woman
<verb>        -> loves | hates | marries | divorces
<modifier>    -> a little | a lot | forever | sometimes
```

Example 1 Parse Tree

- **Example sentence:**
  “man loves woman forever”



Example 2 Production Rules

- for a subset of the German language

```
<sentence>        -> <subject phrase> <verb> <object phrase>
<subject phrase> -> <determiner> <adjective> <noun>
<object phrase>  -> <determiner> <adjective> <noun>

<determiner>     -> der | die | das | den
<noun>           -> Mann | Frau | Kind | Hund | Katze
<verb>           -> mag | schimpft | vergisst|
                    verehrt | verzehrt
<adjective>      -> schoene | starke | laute | duenne
```

Suitability of Production Rules

- **expressiveness**
  - can relevant aspects of the domain knowledge be stated through rules?
- **computational efficiency**
  - are the computations required by the program feasible?
- **easy to understand?**
  - can humans interpret the rules
- **easy to generate?**
  - how difficult is it for humans to construct rules that reflect the domain knowledge

Case Studies
Production Rules

- **sample domains**
  - e.g. theorem proving, determination of prime numbers, distinction of objects (e.g. types of fruit, trees vs. telephone poles, churches vs. houses, animal species)
- **suitability of production rules**
  - basic production rules
    - no salience, certainty factors, arithmetic
  - rules in ES/KBS
    - salience, certainty factors, arithmetic
    - e.g. CLIPS, Jess
  - enhanced rules
    - procedural constructs
      - e.g. loops
    - objects
      - e.g. COOL, Java objects
    - fuzzy logic
      - e.g. FuzzyCLIPS, FuzzyJ

Trees and Telephone Poles

- **distinguish between stick diagrams of trees and telephone poles**
- **expressiveness**
  - is it possible to specify a set of rules that captures the distinction?
- **computational efficiency**
  - are the computations required by the program feasible?
- **easy to understand?**
  - the rules can be phrased in such a way that humans can understand them with moderate effort
- **easy to generate?**
  - may be difficult; the problem is to identify criteria that are common for trees, but not shared with telephone poles

Identification and Generation of Prime Numbers

- **identification: for a given number, determine if it is prime**
- **generation: compute the sequence of prime numbers**
- **expressiveness**
  - it is possible to specify identification as well as generation in rules
- **computational efficiency**
  - reasonable if arithmetic is available, very poor if not
- **easy to understand?**
  - the rules can be formulated in an understandable way
- **easy to generate?**
  - may require a good math background

Advantages of Production Rules

- simple and easy to understand
- straightforward implementation in computers possible
- formal foundations for some variants

Problems with Production Rules

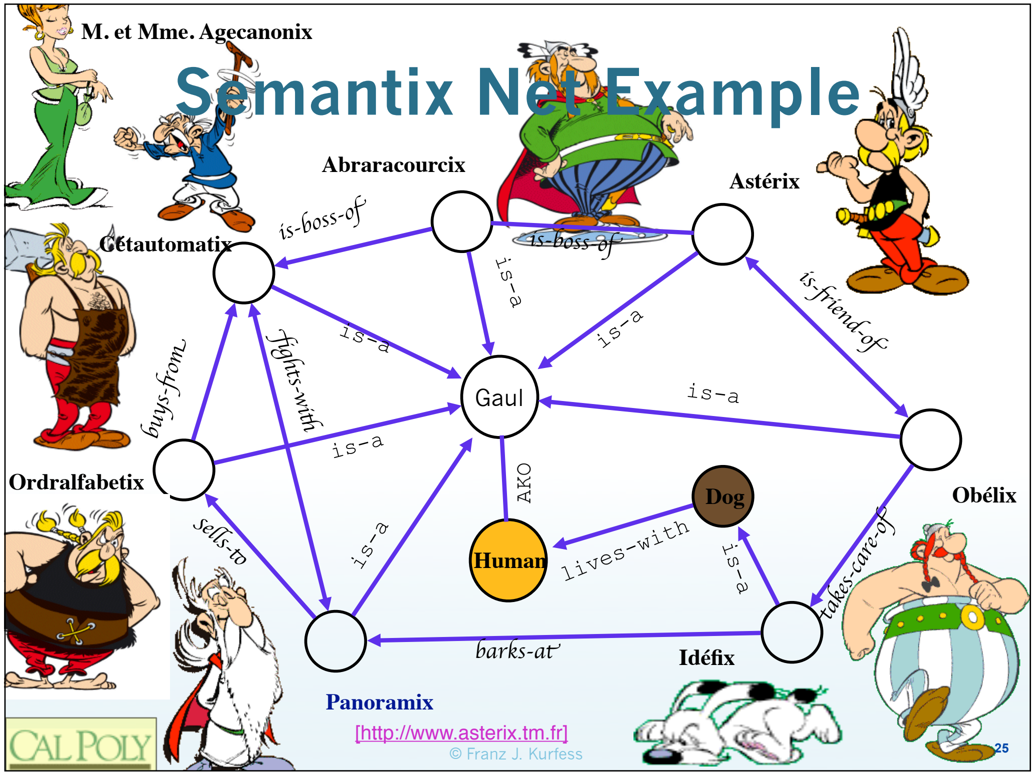
- simple implementations are very inefficient
- some types of knowledge are not easily expressed in such rules
- large sets of rules become difficult to understand and maintain

Semantic Nets

- **graphical representation for propositional information**
- **originally developed by M. R. Quillian as a model for human memory**
- **labeled, directed graph**
- **nodes represent objects, concepts, or situations**
  - labels indicate the name
  - nodes can be instances (individual objects) or classes (generic nodes)
- **links represent relationships**
  - the relationships contain the structural information of the knowledge to be represented
  - the label indicates the type of the relationship

CAL POLY

Semantix Net Example



Semantix Net Cheats

- **colors**
  - should properly be encoded as separate nodes with relationships to the respective objects
- **font types**
  - implies different types of relationships
  - again would require additional nodes and relationships
- **class relationships**
  - not all dogs live with Gauls
  - AKO (a-kind-of) relationship is special (inheritance)
- **instances**
  - arrows from individual humans to the class Human omitted
    - assumes that AKO allows inheritance
- **directionality**
  - the direction of the arrows matters, not that of the text

Relationships

- **without relationships, knowledge is an unrelated collection of facts**
  - reasoning about these facts is not very interesting
    - inductive reasoning is possible
- **relationships express structure in the collection of facts**
  - this allows the generation of meaningful new knowledge
    - generation of new facts
    - generation of new relationships

Types of Relationships

- **relationships can be arbitrarily defined by the knowledge engineer**
  - allows great flexibility
  - for reasoning, the inference mechanism must know how relationships can be used to generate new knowledge
    - inference methods may have to be specified for every relationship
- **frequently used relationships**
  - IS-A
    - relates an instance (individual node) to a class (generic node)
  - AKO (a-kind-of)
    - relates one class (subclass) to another class (superclass)

Objects and Attributes

- **attributes provide more detailed information on nodes in a semantic network**
  - often expressed as *properties*
    - combination of attribute and value
  - attributes can be expressed as relationships
    - e.g. has-attribute

Implementation Questions

- **simple and efficient representation schemes for semantic nets**
  - tables that list all objects and their properties
  - tables or linked lists for relationships
- **conversion into different representation methods**
  - predicate logic
    - nodes correspond variables or constants
    - links correspond to predicates
  - propositional logic
    - nodes and links have to be translated into propositional variables and properly combined with logical connectives

OAV-Triples

- **object-attribute-value triples**
  - can be used to characterize the knowledge in a semantic net
  - quickly leads to huge tables

Object	Attribute	Value
Astérix	profession	warrior
Obélix	size	extra large
Idéfix	size	petite
Panoramix	wisdom	infinite

Problems Semantic Nets

- **expressiveness**
  - no internal structure of nodes
  - relationships between multiple nodes
  - no easy way to represent heuristic information
  - extensions are possible, but cumbersome
  - best suited for binary relationships
- **efficiency**
  - may result in large sets of nodes and links
  - search may lead to combinatorial explosion
    - especially for queries with negative results
- **usability**
  - lack of standards for link types
  - naming of nodes
    - classes, instances

Schemata

- **suitable for the representation of more complex knowledge**
  - causal relationships between a percept or action and its outcome
  - “deeper” knowledge than semantic networks
    - nodes can have an internal structure
  - for humans often tacit knowledge
- **related to the notion of records in computer science**

Concept Schema

- **abstraction that captures general/typical properties of objects**
  - has the most important properties that one usually associates with an object of that type
    - may be dependent on task, context, background and capabilities of the user, ...
  - similar to stereotypes
- **makes reasoning simpler by concentrating on the essential aspects**
- **may still require relationship-specific inference methods**

Schema Examples

- **the most frequently used instances of schemata are**
  - frames [Minsky 1975]
  - scripts [Schank 1977]
- **frames consist of a group of slots and fillers to define a stereotypical objects**
- **scripts are time-ordered sequences of frames**

Frame

- **represents related knowledge about a subject**
  - provides default values for most slots
- **frames are organized hierarchically**
  - allows the use of inheritance
- **knowledge is usually organized according to cause and effect relationships**
  - slots can contain all kinds of items
    - rules, facts, images, video, comments, debugging info, questions, hypotheses, other frames
  - slots can also have procedural attachments
    - procedures that are invoked in specific situations involving a particular slot
      - on creation, modification, removal of the slot value

Simple Frame Example

Slot Name	Filler
name	Astérix
height	small
weight	low
profession	warrior
armor	helmet
intelligence	very high
marital status	presumed single

Overview of Frame Structure

- **two basic elements: slots and facets (fillers, values, etc.);**
- **typically have parent and offspring slots**
  - used to establish a property inheritance hierarchy (e.g., specialization-of)
- **descriptive slots**
  - contain declarative information or data (static knowledge)
- **procedural attachments**
  - contain functions which can direct the reasoning process (dynamic knowledge) (e.g., "activate a certain rule if a value exceeds a given level")
- **data-driven, event-driven (bottom-up reasoning)**
- **expectation-drive or top-down reasoning**
- **pointers to related frames/scripts - can be used to transfer control to a more appropriate frame**

[Rogers 1999]

Usage of Frames

- **filling slots in frames**
  - can inherit the value directly
  - can get a default value
  - these two are relatively inexpensive
  - can derive information through the attached procedures (or methods) that also take advantage of current context (slot-specific heuristics)
  - filling in slots also confirms that frame or script is appropriate for this particular situation

Restaurant Frame Example

- **generic template for restaurants**
  - different types
  - default values
- **script for a typical sequence of activities at a restaurant**

[Rogers 1999]

Generic RESTAURANT Frame

Specialization-of: Business-Establishment

Types:

range: (Cafeteria, Fast-Food, Seat-Yourself, Wait-To-Be-Seated)

default: Seat-Yourself

if-needed: IF plastic-orange-counter THEN Fast-Food,

IF stack-of-trays THEN Cafeteria,

IF wait-for-waitress-sign or reservations-made THEN Wait-To-Be-Seated,

OTHERWISE Seat-Yourself.

Location:

range: an ADDRESS

if-needed: (Look at the MENU)

Name:

if-needed: (Look at the MENU)

Food-Style:

range: (Burgers, Chinese, American, Seafood, French)

default: American

if-added: (Update Alternatives of Restaurant)

Times-of-Operation:

range: a Time-of-Day

default: open evenings except Mondays

Payment-Form:

range: (Cash, CreditCard, Check, Washing-Dishes-Script)

Event-Sequence:

default: Eat-at-Restaurant Script

Alternatives:

range: all restaurants with same Foodstyle

if-needed: (Find all Restaurants with the same Foodstyle)

[Rogers 1999]

Restaurant Script

EAT-AT-RESTAURANT Script

Props: (Restaurant, Money, Food, Menu, Tables, Chairs)

Roles: (Hungry-Persons, Wait-Persons, Chef-Persons)

Point-of-View: Hungry-Persons

Time-of-Occurrence: (Times-of-Operation of Restaurant)

Place-of-Occurrence: (Location of Restaurant)

Event-Sequence:

first: Enter-Restaurant Script

then: if (Wait-To-Be-Seated-Sign or Reservations)
then Get-Maitre-d's-Attention Script

then: Please-Be-Seated Script

then: Order-Food-Script

then: Eat-Food-Script unless (Long-Wait) when Exit-Restaurant-Angry Script

then: if (Food-Quality was better than Palatable)
then Compliments-To-The-Chef Script

then: Pay-For-It-Script

finally: Leave-Restaurant Script

[Rogers 1999]

Frame Advantages

- **fairly intuitive for many applications**
  - similar to human knowledge organization
  - suitable for causal knowledge
  - easier to understand than logic or rules
- **very flexible**

Frame Problems

- **it is tempting to use frames as definitions of concepts**
  - not appropriate because there may be valid instances of a concept that do not fit the stereotype
  - exceptions can be used to overcome this
    - can get very messy
- **inheritance**
  - not all properties of a class stereotype should be propagated to subclasses
  - alteration of slots can have unintended consequences in subclasses

Logistics - Jan 24, 2013

- **Project**
  - Team repositories: TRAC Wiki, alternatives
    - will start grading of
      - project description, background and related work, difficulty, relevance
- **KB Nugget presentations**
  - who's presenting today?
  - Topics
  - Signup for Date & Time Slots
- **Quiz**
  - Quiz 2 available Thu until midnight
- **Assignments**
  - A1: Concept Map
    - due Jan 31

Logic

- **here: emphasis on knowledge representation purposes**
  - logic and reasoning is discussed in the next chapter

Representation, Reasoning and Logic

- **two parts to knowledge representation language:**
  - syntax
    - describes the possible configurations that can constitute sentences
  - semantics
    - determines the facts in the world to which the sentences refer
    - tells us what the agent believes

[Rogers 1999]

Reasoning

- **process of constructing new configurations (sentences) from old ones**
  - proper reasoning ensures that the new configurations represent facts that actually follow from the facts that the old configurations represent
  - this relationship is called **entailment** and can be expressed as
    KB |= alpha
    - knowledge base KB entails the sentence alpha

Inference Methods

- **an inference procedure can do one of two things:**
  - given a knowledge base KB, it can derive new sentences α that are (supposedly) entailed by KB
    KB |- α ==> KB |= α
  - given a knowledge base KB and another sentence alpha, it can report whether or not alpha is entailed by KB
    KB ∧ α ==> KB |= α
- **an inference procedure that generates only entailed sentences is called sound or truth-preserving**
- **the record of operation of a sound inference procedure is called a proof**
- **an inference procedure is complete if it can find a proof for any sentence that is entailed**

KR Languages and Programming Languages

- **how is a knowledge representation language different from a programming language (e.g. Java, C++)?**
  - programming languages can be used to express facts and states
- **what about**
  - "there is a pit in [2,2] or [3,1] (but we don't know for sure)"
  - "there is a wumpus in some square"
- **programming languages are not expressive enough for situations with incomplete information**
  - we only know some possibilities which exist

[Rogers 1999]

KR Languages and Natural Language

- **how is a knowledge representation language different from natural language**
  - e.g. English, Spanish, German, …
- **natural languages are expressive, but have evolved to meet the needs of communication, rather than representation**
- **the meaning of a sentence depends on the sentence itself and on the context in which the sentence was spoken**
  - e.g. "Look!"
- **sharing of knowledge is done without explicit representation of the knowledge itself**
- **ambiguous (e.g. small dogs and cats)**

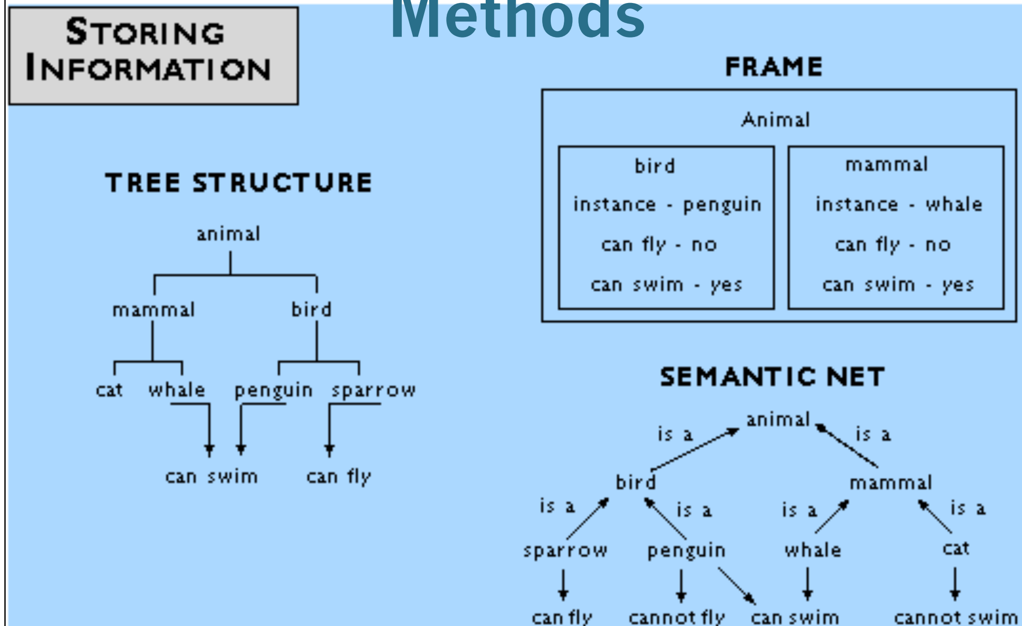
[Rogers 1999]

Good Knowledge Representation Languages

- **combine the best of natural and formal languages:**
  - expressive
  - concise
  - unambiguous
  - independent of context
    - what you say today will still be interpretable tomorrow
  - efficient
    - the knowledge can be represented in a format that is suitable for computers
    - practical inference procedures exist for the chosen format
  - effective
    - there is an inference procedure which can act on it to make new sentences

[Rogers 1999]

Example: Representation Methods



[Guinness 1995]

Resource Description Format (RDF)

Context: Semantic Web

Low-level representation scheme

Implementation

extension of XML

triple stores

Semantic Web

By Kevin McCullough

CSC 581 Dr. Franz Kurfess

What is RDF?

- Foundation for processing metadata
  - provides interoperability between applications that exchange machine-understandable information on the Web
- Arbitrarily expressive language
- Syntax-neutral (although uses XML)

What can you do with RDF?

- Resource discovery
  - better search engine capabilities
- Cataloging
  - describing the content and content relationships of a particular Web site, page, or digital library
- Intelligent software agents
  - to facilitate knowledge sharing and exchange
- Content rating
  - describing collections of pages that represent a single logical "document"
- Describing intellectual property rights of Web pages
- Expressing privacy preferences
  - user as well
  - privacy policies of a Web site

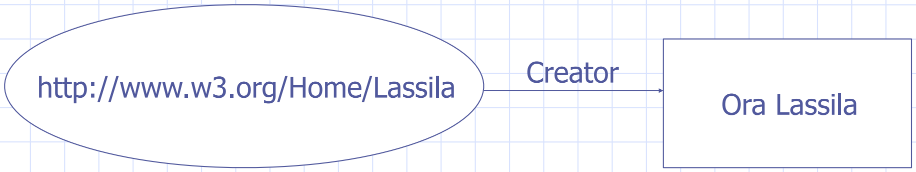
More RDF?

- RDF should be
  - domain neutral
  - evolvable
  - capable of “learning”, mixing previous knowledge and data with that acquired on the semantic web
- RDF does not specify a mechanism for reasoning
  - can be viewed as a simple frame system
  - A reasoning mechanism can be built on top of this frame system

RDF structure example

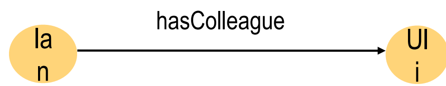
Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila

Subject (Resource) (*record)	http://www.w3.org/Home/Lassila
Predicate (Property) (*column)	Creator
Object (literal) (*cell)	"Ora Lassila"



The RDF Data Model

- Statements are <subject, predicate, object> triples:
  `<Ian,hasColleague,Uli>`
- Can be represented as a graph:



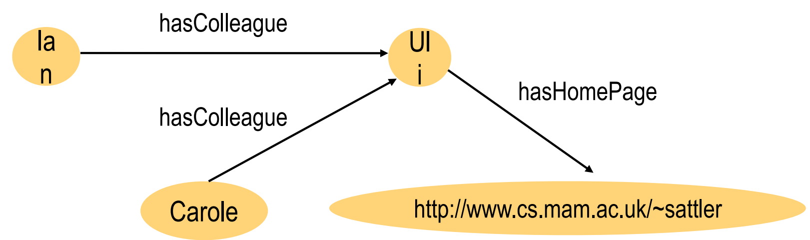
- Statements describe properties of resources
- A resource is any object that can be pointed to by a URI:
  - a document, a picture, a paragraph on the Web;
  - http://www.cs.man.ac.uk/index.html
  - a book in the library, a real person (?)
  - isbn://5031-4444-3333
  - …
- Properties themselves are also resources (URIs)

URIs

- URI = Uniform Resource Identifier
- "The generic set of all names/addresses that are short strings that refer to resources"
- URLs (Uniform Resource Locators) are a particular type of URI, used for resources that can be accessed on the WWW (e.g., web pages)
- In RDF, URIs typically look like “normal” URLs, often with fragment identifiers to point at specific parts of a document:
  - http://www.somedomain.com/some/path/to/file#fragmentID

Linking Statements

- The subject of one statement can be the object of another
- Such collections of statements form a directed, labeled graph



- Note that the object of a triple can also be a “literal” (a string)

RDF Syntax

- RDF has an XML syntax
- Every `Description` element describes a resource
- Every attribute or nested element inside a `Description` is a `property` of that Resource
- We can refer to resources by using URIs

```
<Description about="some.uri/person/ian_horrocks">
   <hasColleague resource="some.uri/person/uli_sattler"/>
</Description>
<Description about="some.uri/person/uli_sattler">
   <hasHomePage>http://www.cs.mam.ac.uk/~sattler</hasHomePage>
</Description>
<Description about="some.uri/person/carole_goble">
   <hasColleague resource="some.uri/person/uli_sattler"/>
</Description>
```

RDF Schema (RDFS)

- RDF gives a language for meta data annotation, and a way to write it down in XML, but it does not provide any way to structure the annotations
- RDF Schema augments RDF to allow you to define vocabulary terms and the relations between those terms
  - it gives "extra meaning" to particular RDF predicates and resources
  - e.g., Class, subClassOf, domain, range
- These terms are the RDF Schema building blocks (constructors) used to create vocabularies:

```
<Person,type,Class>
<hasColleague,type,Property>
<hasColleague,range,Person>
<hasColleague,domain,Person>
<Professor,subClassOf,Person>
<Carole,type,Professor>
<Carole,hasColleague,Ian>
```

RDF and RDFS circa 2001

- Initial definition of RDF and RDFS was informal
  - A document giving an English description of what everything meant
  - Not adequate for representation
    - Debate on exact meaning of constructs, e.g., blank nodes
  - Similar to problems with informal Knowledge Representation work
- W3C chartered the RDF Core Working Group to fix this (and other problems)
  - Produced cleaned up syntax for RDF
  - Produced formal semantics
- RDF and RDFS are now real representation languages
  - Formal syntax, formal semantics, inference

Repositories

Repository Types

- **Index**
  - list of occurrences of strings that point to the original documents
- **file system**
- **(relational) data base**
  - set of records based on one or more tables
- **non-relational data base (No-SQL)**
  - more flexible internal structure
  - often used for Web-scale shallow objects
  - examples: Hadoop, Hbase, Cassandra
- **RDF repository (triple store)**
  - low-level storage facility
  - relies on simple statements that connect two entities through a relation
    - object, attribute, value,
    - subject, predicate, object

Repository Examples

- Wikipedia
- dbpedia
- Freebase
- Cyc

RDF Repositories

- also known as “triple stores”
- often combined with ontology managers
- see W3C Web site for an overview
  - http://www.w3.org/wiki/LargeTripleStores
  - http://www.w3.org/wiki/SemanticWebTools#RDF_Triple_Store_Systems
- openRDF.org
- OpenLink Virtuoso
- BigOWLIM
- Allegrograph Franz Inc
- Oracle Spatial 11g

Triple Store Evaluation

- **functional evaluation**
- **performance evaluation**
- **see Triple Store Evaluation Analysis Report by Revelytix, Inc.**
  - http://www.revelytix.com/sites/default/files/TripleStoreEvaluationAnalysisResults.pdf
    - report states "Confidential, do not distribute without permission of Revelytix" but is available on their Web site

Ontologies

- **principles**
  - definition of terms
    - lexicon, glossary
  - relationships between terms
    - taxonomy, thesaurus
- **purpose**
  - establishing a common vocabulary for a domain
- **graphical representation**
  - UML, topic maps,
- **examples**
  - IEEE SUO, SUMO, Cyc, WordNet

Terminology

- **ontology**
  - provides semantics for concepts
  - words are used as descriptors for concepts
- **lexicon**
  - provides semantics for all words in a language by defining words through descriptions of their meanings
- **thesaurus**
  - establishes relationships between words
    - synonyms, homonyms, antonyms, etc.
  - often combined with a taxonomy
- **taxonomy**
  - hierarchical arrangement of concepts
  - often used as a “backbone” for an ontology

What is the Semantic Web?

- **Based on the World Wide Web**
- **Characterized by resources, not text and images**
  - Meant for software agents, not human viewers
  - Defined by structured documents that reference each other, forming potentially very large networks
  - Used to simulate knowledge in computer systems
- **Semantic Web documents can describe just about anything humans can communicate about**

Ontologies and the Semantic Web

- **Ontologies are large vocabularies**
  - Defined within Semantic Web documents (OWL)
  - Define languages for other documents (RDF)
  - Resources can be instances of ontology classes
- **Upper Ontologies define basic, abstract concepts**
- **Lower Ontologies define domain-specific concepts**
- **Meta-ontologies define ontologies themselves**

Ontology Terms

- **precision**
  - a term identifies exactly one concept
- **expressiveness**
  - the representation language allows the formulation of very flexible statements
- **descriptors for concepts**
  - ideally, there should be a one-to-one mapping between a term and the associated concept (and vice versa): high precision, and high expressiveness
    - this is not the case for natural languages
    - “parasitic interpretation” of terms often implies meaning that is not necessarily specified in the ontology

IEEE Standard Upper Ontology

- **project to develop a standard for ontology specification and registration**
- **based on contributions of three SUO candidate projects**
  - IFF
  - OpenCyc/CycL
  - SUMO

Standard Upper Ontology Working Group (SUO WG), Cumulative Resolutions, 2003, http://suo.ieee.org/SUO/resolutions.html

OpenCyc

- **derived from the development of Cyc**
  - a very large-scale knowledge based system
- **Cycorp, The Syntax of CycL, 2002, http://www.cyc.com/cycdoc/ref/cycl-syntax.html**

SUMO

- stands for "Suggested Upper Merged Ontology"
- Niles, Ian, and Adam Pease, Towards a Standard Upper Ontology, 2001
- Standard Upper Ontology Working Group (SUO WG), Cumulative Resolutions, 2003, http://suo.ieee.org/SUO/resolutions.html

WordNet

- online lexical reference system
  - design is inspired by current psycholinguistic theories of human lexical memory
- English nouns, verbs, adjectives and adverbs
  - organized into synonym sets, each representing one underlying lexical concept
- related efforts for other languages

Lojban

- **artificial, logical, human language derived from a language called Loglan**
- **one-to-one correspondence between concepts and words**
  - high precision
- **high expressiveness**
- **audio-visually isomorphic nature**
  - only one way to write a spoken sentence
  - only one way to read a written sentence
- **Logical Language Group, Official Baseline Statement, 2005**
  - http://www.lojban.org/llg/baseline.html

What is Lojban?

- **A constructed/artificial language**
- **Developed from Loglan**
  - Dr. James Cooke Brown
  - Introduced between 1955-1960
- **Maintained by The Logical Language Group**
  - also known as *la lojbangirz*
  - branched Lojban off from Loglan in 1987

[Brandon Wirick, 2005]

Main Features of Lojban

- Usable by Humans and Computers
- Culturally Neutral
- Based on Logic
- Unambiguous but Flexible
- Phonetic Spelling
- Easy to Learn
- Large Vocabulary
- No Exceptions
- Fosters Clear Thought
- Variety of Uses
- Demonstrated with Prose and Poetry
- Allows Jokes ;-}

[Brandon Wirick, 2005]

Lojban at a Glance

Example sentence in English: “Wild dogs bite.”

Translation into Lojban: “loi cicyge'u cu batci”

cilce (cic) - x1 is wild/untamed

gerku (ger, ge'u) - x1 is a dog/canine of species/breed x2

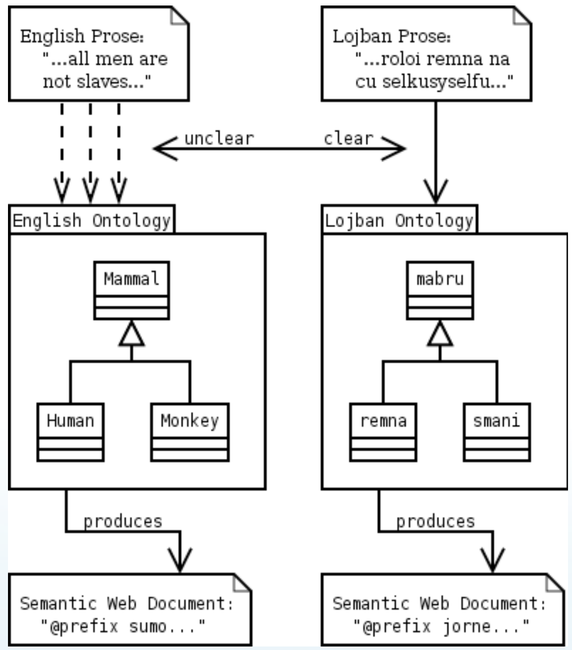
batci (bat) - x1 bites/pinches x2 on/at specific locus x3 with x4

cilce gerku → (cic) (ge'u) → cicyge'u

Lojban and the Semantic Web

- **Currently, most upper ontologies use English**
  - Not really English, but arbitrary class names
  - Classes' meanings cannot be directly inferred from their names, nor vice-versa
- **Translating English prose into Semantic Web documents can be difficult**
  - Class choices depend on context within prose
  - English prose is highly idiomatic
- **Lojban does not have these problems**

[Brandon Wirick, 2005]



[Brandon Wirick, 2005]

OWL to the Rescue

- **XML-based**
  - "RDF on steroids"
- **Designed for inferencing**
- **Closer to the domain**
- **Don't need a PhD to understand it**
- **Information sharing**
  - RDF-compatible because it is RDF
  - Growing number of published OWL ontologies
  - URIs make it easy to merge equivalent nodes
- **Different levels**
  - OWL lite
  - OWL DL (description logics)
  - OWL full (predicate logic)

[Frank Vasquez, 2005]

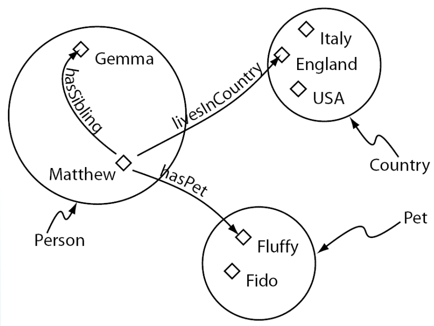
Description Logic

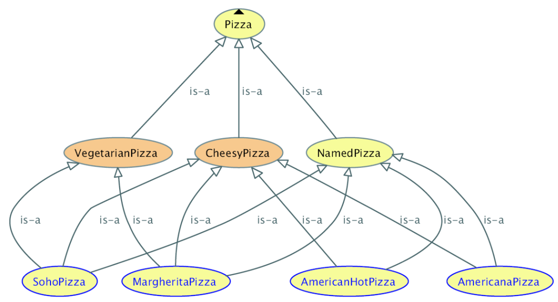
- **Classes**
  - Things, categories, concepts.
  - Inheritance hierarchies via subclasses.
- **Properties**
  - Relationships, predicates, statements.
  - Can have subproperties.
- **Individuals**
  - Instances of a class.
  - Real subjects and objects of a predicate.

[Frank Vasquez, 2005]

Visualizing the Data Model

- Venn Diagrams and Semantic Networks.



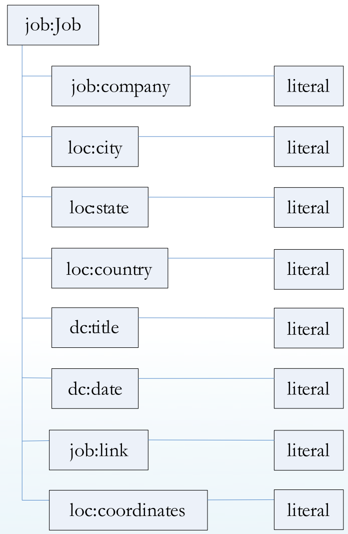


Images from University of Manchester

RDF Ontologies

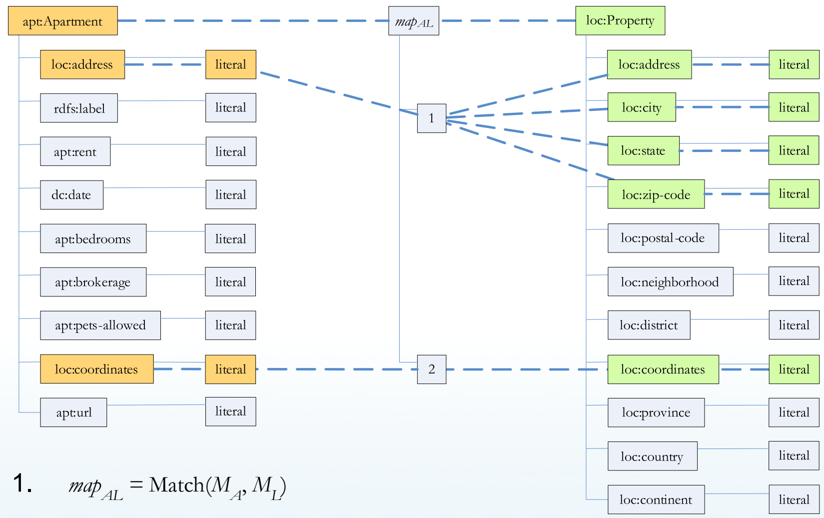
- Dublin Core
- FOAF
- RDF vCard
- RDF Calendar

- SIMILE Location
- SIMILE Job
- SIMILE Apartment



[Frank Vasquez, 2005]

Fixing Modeling Conflicts



1. $map_{AL} = \text{Match}(M_A, M_L)$

Important Concepts and Terms

- attribute
- common-sense knowledge
- concept
- data
- derivation
- entailment
- epistemology
- expert system (ES)
- expert system shell
- facet
- frame
- graph
- If-Then rules
- inference
- inference mechanism
- information
- knowledge
- knowledge base
- knowledge-based system
- knowledge representation
- link
- logic
- meta-knowledge
- node
- noise
- object
- production rules
- reasoning
- relationship
- rule
- schema
- script
- semantic net
- slot

Summary Knowledge Representation

- **knowledge representation is very important for knowledge-based system**
- **popular knowledge representation schemes are**
  - rules, semantic nets, schemata (frames, scripts), logic
- **the selected knowledge representation scheme should have appropriate inference methods to allow reasoning**
- **a balance must be found between**
  - effective representation, efficiency, understandability
- **the Semantic Web incorporates powerful KR techniques**
  - Web Ontology Language (OWL) for ontology descriptions
  - Resource Description Format (RDF) for knowledge repositories