Ontology

Assoc. Prof. Ian Watson
CS 367

Contents

Definition

- Ontology
  - (1): a science or study of being; specifically, a branch of metaphysics relating to the nature and relations of being. (2): a particular system according to which problems of the nature of being are investigated.
  - a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system.

Webster's 3rd. New International Dictionary

Definition

- Ontology
  - (1): a science or study of being; specifically, a branch of metaphysics relating to the nature and relations of being. (2): a particular system according to which problems of the nature of being are investigated.
  - a theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system.
- language -> communication & understanding

Why?

- If people or agents are to communicate they must share a common understanding
- “Waiter, that was a beautiful duck, please get me the bill”
- The elephant stood on the table, it broke!

Why?

- Understanding means we share the meaning of words or concepts
  - Red, tiger, ocean, love, mother…
- Understanding also needs common sense
  - Which is quicker, a *fast* car or a *fast* plane?

Why?

- Ontologies deal with defining concepts and relations
- Conceptual graphs, KIF, Ontolingua, commonKADS, ...
- They are a dictionary
- They can also be used to define problem solving K and common sense K

Why?

- An ontology is a fomal description of the concepts and relations shared by a community of agents
- Like a formal specification of a program

An example

- Mother
  - An animal that is the female parent of children
- Animal
  - A living thing – contains DNA
- Living
  - A temporary state for some things requiring energy
- Thing
  - An atomic concept

An example

- Mother from the CYC ontology
  - #$biologicalMother : <#$Animal><#$FemaleAnimal> (#$biologicalMother OFFSPRING FEMALE) means that #$FemaleAnimal FEMALE is the female biological parent of the #$Animal OFFSPRING.

An example

- #$Animal
  - The collection of all animals; this large class of organisms is one instance of #$BiologicalKingdom. Animals are typically motile, living, whole organisms; they are elements of #$Heterotroph, incapable of performing instances of #$Photosynthesis. Animal cells contain cholesterol and lack cell walls made of cellulose. #$Person is a subset of #$Animal

A problem

- These definitions necessarily bottom out in expressions containing undefinable primitive atomic concepts
- We provide the *meaning*
- This is *The Chinese Room* problem identified by Searle
- The computer does not *understand* the symbols it manipulates

The University of Auckland

The Knowledge Level

- Allen Newell (1982)
  AI Vol. 18 pp. 87-127

abstraction

- Knowledge Level (goals, actions...)
- Program (symbol) Level (data, commands)
- Logic Level (bits)
- Circuit Level (current, voltage)
- Device Level (electrons)

to write a program you do not need to instruct electrons where to go

The University of Auckland

The Knowledge Level

- To describe K you do not need to use a specific programming language

Each level can be reduced to the one below

- Knowledge Level (goals, actions...)
- Program (symbol) Level (data, commands)
- Logic Level (bits)
- Circuit Level (current, voltage)
- Device Level (electrons)

But we do not need to worry how it is implemented

The University of Auckland

The Knowledge Level

- Newell's principle of rationality:
  "*if an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action*"
- A direct relation to autonomous intelligent agents
- Thus the goals and K of agents are described at the Knowledge Level

16

The Knowledge Level

- Knowledge:
  "*whatever can be ascribed to an agent, such that it's behavior can be computed according to the principle of rationality*"
- K is characterised functionally in terms of what an agent does, not how it is represented (encapsulation)

17

The Knowledge Level

- We require a K Level representation to describe the K and goals of our agents
- Allow our agents to communicate
- And perform according to the principle of rationality

18

Declarative Programming?

- AI programs commonly contain K written as statements of fact
  - conventional programs describe procedures for manipulating data (for, until, while…)
- In Prolog
  - mother(X,Y):- female(X), parent(X,Y).
  - X is the Mother of Y if X is Female and X is the Parent of Y.

Declarative Programming

- Prolog (Progamming in Logic) is an implementation of Predicate Calculus
- all basket ball players are tall

$$\forall X\,(\text{basketball\_player}(X) \Rightarrow X(\text{tall}))$$

- some people like brussel sprouts

$$\exists X(\text{person}(X) \Rightarrow \text{likes}(X, \text{sprouts}))$$

Declarative Programming

- Why not use Logic as our ontological K level language?
- logic is hard to read – even for computer scientists
- Some statements are hard to express
  - "everyone loves someone"
- Poor communication medium

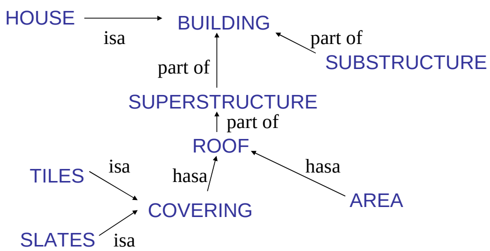
K & Communication

- Which is easier to understand?
  - isa(house,building).
    isa(slates,covering).
    isa(tiles,covering).
    partof(substructure,building).
    partof(superstructure,building).
    partof(roof,superstructure).
    hasa(covering,roof).
    hasa(area,roof).

The University of Auckland

K & Communication

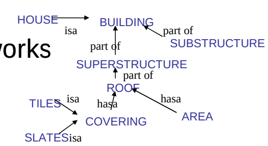
- Which is easier to understand?



The University of Auckland

K & Communication

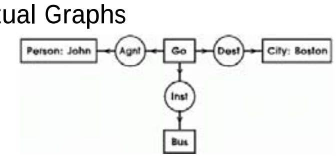
- Prolog can be represented graphically
- Prolog is logic
- Therfore logic can be represented graphically
- As Semantic networks



The University of Auckland

K & Communication

- Semantic networks range from ad hoc partially formalised representations using simple labels (isa, hasa, partof)
- To formal representations such as Conceptual Graphs



KIF

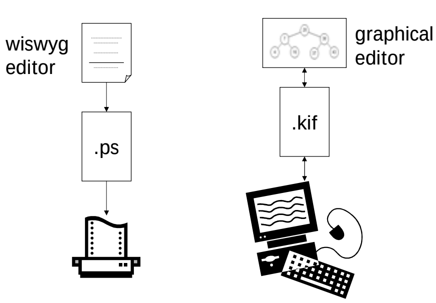
- The Knowledge Interchange Format
  - A computer-oriented language for the interchange of knowledge among disparate programs

http://www-ksl.stanford.edu/knowledge-sharing/kif/

KIF

- Not intended as a primary means of communication with people
- Intended to underpin other representations (graphical, NL, etc)
- Not intended as a direct computational rep.
- Analogous to PostScript as a document description language

KIF analgous to .ps



KIF features

- Declarative semantics – but does not require a specific interpreter (eg unlike Prolog)
- Logically comprehensive – not resricted to Horn clauses like Prolog
- Handles meta-knowledge – allows for explicit K-representations

KIF features

- Translatablity – supports the translation to and from different K-reps
- Readability – not a primary feature but it can be read by people
- Usability – not a primary feature but it can be implemented computationally

KIF syntax

- Overview – look to KIF Manual for details
- 2 forms
  - Linear form (ASCII strings)
  - Structured form (objects)
- Inherited its syntax from LISP

KIF syntax

- *word* is a KIF primitive
  - <word> ::= *a primitive syntatic object*
- *expression* is a word or a finite sequence of expressions
  - <expression> ::= <word> | (<expression>*)
- KIF defines variables, constants operators and relations based upon primitives

KIF syntax

- 4 special types of expression
- Terms
  - Describe objects in the world being described
- Sentences
  - Express facts about the world
- Rules
  - Express inference steps
- Definitions
  - Define constants

KIF syntax

- Forms are sets of terms, sentences, rules and definitions
- A set of forms comprises a knowledge-base
- The set is not ordered
- There is no sequence
  - (declarative not procedural)

34

KIF conceptualisation

- These objects occur in <u>all</u> KIF universes
- *words* – KIF words are basic objects
- All complex numbers
- All finite sets of objects in the universe
- $\perp$ pronounced “bottom” a special object used where no further meaning can be derived

35

KIF semantics

- KIF is a formally defined language
- fairly complex semantics
- Look at the KIF manual for a full definition (resources section of course website)

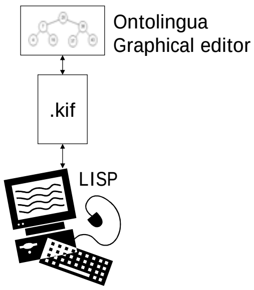
36

Ontolingua

- Ontolingua provides a distributed collaborative environment to browse, create, edit, modify, and use ontologies
- The Ontolingua server supports hundreds of users
- An application on top of KIF

www.ksl.stanford.edu/software/ontolingua/

Ontolingua



Ontolingua

- Why develop an ontology
  - To enable a machine to use the knowledge in some application.
  - To enable multiple machines (agents) to share their knowledge.
  - To help yourself understand some area of knowledge better.
  - To help other people understand some area of knowledge.
  - To help people reach a consensus in their understanding of some area of knowledge.

Ontolingua – design method

- Describe the general subject area of your ontology, including any simplifying assumptions you are making
- List what you would like to state in the ontology
- List the concepts that you think should be included in the ontology.
- Look for ontologies in the library that may contain terms which you can use to develop your ontology.
- Review and make modifications to your lists as needed throughout these steps.

Ontolingua – tutorial

- A good online tutorial is available at:

www.ksl.stanford.edu/software/ontolingua/

  - Develops an ontology for used car (vehicle) sales
  - Shows that a new ontology can build on previous ontologies – reuse
  - Guides you through the creation processes

Ontolingua – users

- Ontologies for e-commerce – related to the semantic web initiative
  - Goal is to create a machine readable common language for information exchange on the web
  - Enable applications to understand each others terms
  - Eg web search for "*football*" is that NZ rugby football, US football, or global football (i.e. soccer)

www.w3.org/DesignIssues/Semantic.html

Ontolingua – users

- Ontologies for business process modelling
  - Goal is to create a common language for descison making in business

www.aiai.ed.ac.uk/~entprise/

Ontolingua – users

- Medical ontologies
- Military ontologies
- Academic ontologies

Cyc

- A huge ontology
- Over 1,000,000 assertions (rules)
- Handbuilt over 15 years
- Particular emphasis on “common sense” knowledge
- Run by Doug Lenat an AI pioneer

http://www.cyc.com/

Cyc

- Cyc is a commercial product
- The Upper-Cyc ontology is in the public domain
- ~5,000 terms are defined
- Represent the most common terms in the human perceptual universe
- Maps to CycL a 1st order predicate calculus

46

Cyc

- The authors claim that Cyc is:
  - *Universal* – any concept real or imaginary can be found an appropriate place in the Cyc ontology
  - *Articulate* – distinctions within the ontology between concepts are both necessary and sufficient

47

Cyc syntax

- For each concept Cyc lists
  - The Cyc name for the concept
  - an English comment on the intended meaning and use of the concept
  - a few of the taxonomic "links" which Cyc uses to hierarchically order and interconnect its concepts

48

Cyc syntax

- each concept is represented by a Cyc *term* (denoted by #$Term)
- The Cyc name for the concept
- an English comment on the intended meaning and use of the concept
- a few of the taxonomic "links" which Cyc uses to hierarchically order and interconnect its concepts

Cyc syntax

#$Skin

A (piece of) skin serves as outer protective and tactile sensory covering for (part of) an animal's body. This is the collection of all pieces of skin. Some examples include #$TheGoldenFleece (representing an entire skin of an animal) and (#$BodyPartFn #$YulBrynner #$Scalp) (representing a small portion of his skin).
isa: #$AnimalBodyPartType
genls: #$BiologicalLivingObject #$AnimalBodyPart #$SheetOfSomeStuff #$VibrationThroughAMediumSensor #$TactileSensor

Cyc syntax

- Collections
  - a concept representing a set or class of things with some properties in common,
  - generally what is thought of as "a natural kind."
  - #$Skin is a collection – the set of all full or partial skins

Cyc syntax

- Cyc Relations (predicates and functions)

#$mother : <Animal> <FemaleAnimal>
(#$mother ANIM FEM) means that the #$FemaleAnimal FEM is the female biological parent of the #$Animal ANIM.
isa: #$FamilyRelationSlot #$BinaryPredicate

- #$mother is a relation (a predicate) (notice lower case “m”)
- <Animal> is an arguement

Cyc syntax

- Cyc Relations (predicates and functions)
- Functions are similar except they return
  - True or False
  - a term
  - or a collection

Cyc basic vocabulary

#$Thing
#$Thing is the universal set: the collection of everything! Every Cyc constant in the Knowledge Base is a member of this collection; in the prefix notation of the language CycL, we express that fact as (#$isa CONST #$Thing). Thus, too, every collection in the Knowledge Base is a subset of the collection #$Thing; in CycL, we express that fact as (#$genls COL #$Thing). See #$isa and #$genls for further explanation of those relationships. Note: There are even a few collections, such as #$CharacterString and #$Integer, which have a #$defnSufficient that recognizes non-constants (such as strings and numbers) as instances of #$Thing.
isa: #$Collection
some subsets: #$Path-Generic #$Intangible #$Individual #$SimpleSegmentOfPath #$Path-Simple #$MathematicalOrComputationalThing #$IntangibleIndividual #$Product #$TemporalThing #$SpatialThing #$Situation #$EdgeOnObject #$FlowPath #$ComputationalObject #$Microtheory (plus 1488 more public subsets, 13568 unpublished subsets)

Upper-Cyc covers

- Time and Dates
- Spatial Relations
- Quantities
- Mathematics
- Contexts
- Groups
- "Doing"
- Transformations
- Changes Of State
- Transfer Of Possession
- Movement
- Parts of Objects
- Composition of Substances
- Agents
- Organizations
- Actors
- Roles
- Professions
- Emotion
- Social
- Biology
- Chemistry
- Physiology
- General Medicine
- Materials
- Waves
- Devices
- Construction
- Financial
- Food
- Clothing
- Weather
- Geography
- Transportation
- Information
- Perception
- Agreements
- Linguistic Terms