OKBC
A Protocol For
Knowledge Base Interoperation

Adam Farquhar
Richard Fikes
James Rice
Vinay Chaudhri
Peter Karp

Exchanging Knowledge - Problems

- Knowledge Representation Systems (KRS) vary:
  - » Different levels of expressiveness
  - » Different degrees of reification
  - » Different inference capabilities
- Knowledge represented in one KRS has not been usable in another
- Tools that work for one KRS will not work for others
- Effort is wasted!

Exchanging Knowledge - Approaches

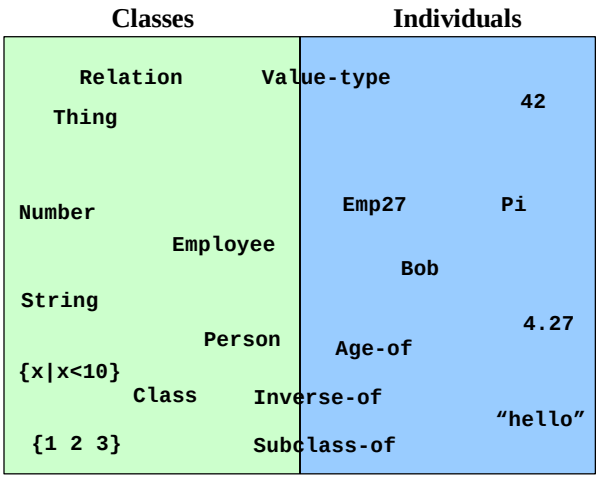
- A Knowledge Interchange Format
  - » KIF specifies a declarative method for exchange
  - » KIF does not address the procedural aspects
    - – Open a KB, Save a KB, Create a class, Delete a class, ...
- An open API
  - » OKBC specifies a protocol for KRS interoperation
  - » OKBC supports a client-server model for interaction
  - » OKBC provides transparent network access
  - » OKBC provides an object-oriented view of a KRS

KRS Variation - Expressiveness

- Ocelot
  - » A simple frame system with classes, individuals, slots, a limited number of facets
- CLIPS
  - » An object system with production rules
- Classic
  - » An elegant classifier system with a substantial set of constraints (e.g., value-type, cardinality)
- Loom
  - » An expressive classifier system with a full assertion language
- ATP
  - » A full FOL theorem prover with limited axiom schema
- Ontolingua
  - » A full KIF representation system

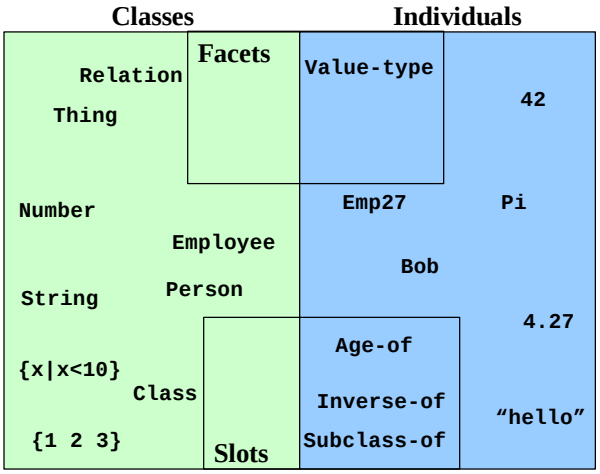
Knowledge Model

- Universe of discourse
- Partition into
  - » Classes
  - » Individuals



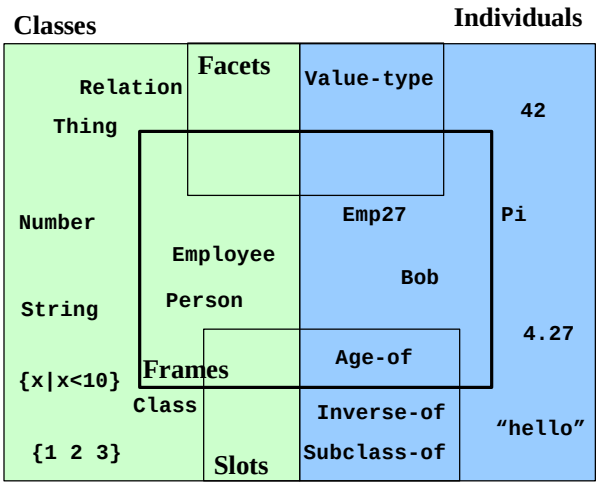
Knowledge Model

- Relationships
  - » Slots
  - » Facets



KRS Design

- KRS distinguish a set of frames
  - » Objects about which assertions may be made
- What constitutes a frame varies widely!
- Restrictions are arbitrary



Dealing with Variation

- What should these OKBC operations return?
  - » Get-kb-frames
  - » Get-kb-classes
  - » Get-kb-individuals
  - » Get-kb-slots
  - » Get-kb-facets
- What guarantees can OKBC provide?
  - » Frames _ classes
  - » Frames _ slots
- Under what conditions could guarantees be stronger?

Dealing with Variation - Names

- KRS use different names for concepts
  - » The most general class: thing, object, all, any, …
  - » Value restriction: value-type, slot-value-type, type, …
- Applications must portably refer to these common objects
- OKBC defines standard names
  - » :thing, :value-type, :inverse, :cardinality, …
  - » If coerce-to-frame(:thing) returns a frame, then it must have the correct meaning
  - » A KRS need not provide any standard names

Dealing with Variation - Inference

- KRS vary widely in inference ability
  - » Simple lookup
  - » Taxonomic reasoning
  - » Full FOL theorem proving
  - » Forward chaining
- Control of inference
  - » Specify inference level: direct, taxonomic, all-inferable
  - » KRSs must return *at least* those values
- Understanding results
  - » Operations return additional information
  - » If the answer is complete
    - – All Universities
    - – All Universities I had time to find
    - – All Universities I can find
  - » If exactly the requested inferences were performed

Dealing with Variation - Types

- Procedural tests allow clients to be more robust
  - » If age-of is a slot on human, then slot-p(age-of) returns True
    - – Facets on age-of for human can be asserted
  - » If frame-p(age-of) returns True, then age-of is also a frame
    - – Properties of age-of can be asserted
- OKBC cannot legislate that all slots are frames