

Ontology

- ▶ A specification of a conceptualization or a set of knowledge terms for a particular domain, including
 - ▶ The vocabulary: concepts and relationships
 - ▶ The semantic interconnections: relationships among concepts and relationships
 - ▶ Some simple rules of inference and logic
- ▶ Some representation languages for ontologies:
 - ▶ Unified Modeling Language (UML) Class Diagrams
 - ▶ Resource Description Framework Language Schema (RDFS)
 - ▶ Web Ontology Language (OWL)
- ▶ Some ontology editors: Protégé, Eddy (OBDA Systems)

Common Ontology

A shared representation between interoperating parties

- ▶ Analogy with humans: shared physical, biological, and social worlds
- ▶ Who builds a common ontology?
- ▶ Standardization efforts, e.g., by IEEE
 - ▶ *Upper-level* ontologies include concepts (of space, time, personhood), which apply across domains
 - ▶ Domain-level ontologies
- ▶ Language-based efforts
 - ▶ WordNet (Princeton)
 - ▶ Longman's Dictionary of Contemporary English
- ▶ Bottom-up approaches build consensus incrementally

Dimensions of Description for Interoperation

- ▶ Descriptions of services that have a bearing on interoperation
- ▶ Constraints that we must discover, represent, and reason about

The Data Dimension

Assumptions about a service's data inputs and outputs

- ▶ Domain specifications
- ▶ Value ranges, e.g., Price ≥ 0
- ▶ Allow or disallow null values

The Structure Dimension

Conceptual model of the information processed by a service

- ▶ Specializations and generalizations of domain concepts
 - ▶ Taxonomic representations and relationships such as in schemas and views, e.g., securities are stocks
- ▶ Semantics of data
 - ▶ Some stock price databases consider daily averages; others closing prices
 - ▶ Value maps, e.g., S&P A+ rating corresponds to Moody's A rating
- ▶ Integrity constraints
 - ▶ Each stock must have a unique SEC identifier
 - ▶ Cardinality constraints

The Process Dimension

Model of operational interactions of a service

- ▶ Procedures, i.e., how to process information, e.g., how to decide what stock to recommend
 - ▶ Flow, e.g., where to forward requests or results
 - ▶ Temporal constraints, e.g., report tax data every quarter
- ▶ Preferences for accessing and updating information
 - ▶ Dealing with data replication (recency or accuracy)
 - ▶ Updating views
- ▶ Contingency strategies (failure recovery), e.g., whether to ignore, redo, or compensate
 - ▶ Contingency procedures, i.e., how to compensate transactions

The Governance Dimension

Model of organizational interactions of a service

- ▶ Privileges, i.e., who has rights to what service
 - ▶ Owners can access all of their accounts, except blind trusts
- ▶ Authentication, i.e., how to establish identity
 - ▶ Passwords, retinal scans, or smart cards
- ▶ Bookkeeping
 - ▶ Logging all accesses

Value Maps

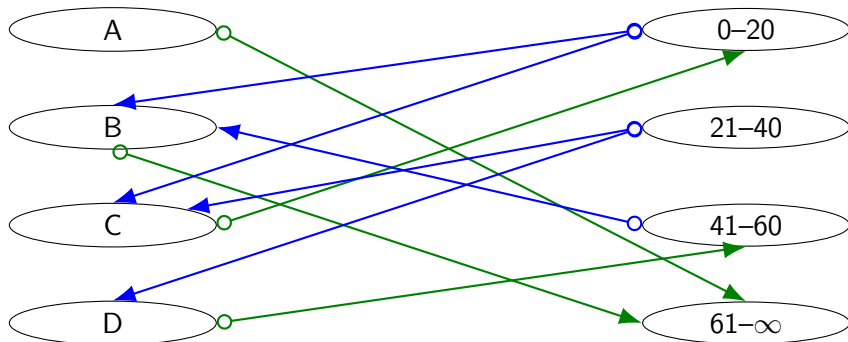
A value map relates the values expressed by different services

- ▶ Two sets of values
- ▶ A function in each direction
- ▶ Important properties satisfied by a “reasonable” value map
 - ▶ Totality or coverage: each function is total
 - ▶ Consistent inversion (à la triple negation): the two functions are adjoint
- ▶ Often each set is ordered in some domain-dependent way; then add
 - ▶ Order preservation or monotonicity: each function preserves order

Example Value Map

Birthday gift selection services

- ▶ User rankings (best to worst): A, B, C, D
- ▶ Prices (in \$): 0–20, 21–40, 41–60, 61–∞



Procedural versus Declarative Knowledge Representation

How versus What

- ▶ Declarative pros: enables standardization, optimization, improved productivity of developers
- ▶ Declarative cons: nontrivial to achieve and causes short-term loss of performance
- ▶ Trade-offs shifted by Web to favor declarative modeling
 - ▶ Because of heterogeneity

Frames versus Descriptions

How to marry intuitive ease and rigor

- ▶ Frame-based approaches are intuitive
 - ▶ Like object-oriented modeling
 - ▶ Rely on names of classes and properties to indicate meaning
- ▶ Description logics provide a computationally rigorous means to represent meaning; difficult for people

Evaluating a Knowledge Representation Language

Expressive power

- ▶ Is unambiguous
- ▶ Supports legitimate inferences
- ▶ Avoids unjustified inferences
- ▶ Reduces reliance on names for meaning
- ▶ Supports representing partial knowledge
- ▶ Supports inserting knowledge incrementally

Compare for Expressiveness and Flexibility

Think of a database in which you store such facts

- ▶ `awg22SolidBlueWire(ID5)`
- ▶ `blueWire(ID5, AWG22, Solid)`
- ▶ `solidWire(ID5, AWG22, Blue)`
- ▶ `wire(ID5, AWG22, Solid, Blue)`
- ▶ `wire(ID5) \wedge size(ID5, AWG22) \wedge type(ID5, solid) \wedge color(ID5, Blue)`

Summary

- ▶ Based on shared ontologies or conceptualizations
 - ▶ Shared models are essential for interoperation
 - ▶ Declarative representations facilitate reasoning about and managing models
 - ▶ Formalization enables ensuring correctness of models and using them for interoperation
- ▶ Good models must accommodate several important considerations
- ▶ Modeling requires several subtle considerations