It Is What It Does: The Pragmatics of Ontology as Language, Contract, and Content

Tom Gruber
Founder and CTO, Intraspect Software
Formerly at Stanford University
What is this talk about?

- What are ontologies?
  - a theoretical framework for “Sharing Experiences and Best Practices”
- What are they for?
  - ways of using ontology, and why they matter
- How do we build them?
  - argument for a design perspective and engineering methodology
Why Create Ontologies?

- to enable data exchange among programs
- to simplify unification (or translation) of disparate representations
- to employ knowledge-based services
- to embody the representation of a theory
- to facilitate communication among people
What is an ontology?

- An ontology is an explicit specification of a conceptualization.
- A conceptualization is an abstract, simplified view of the world that we want to represent.
- If the specification medium is a formal representation, the ontology defines the vocabulary.
Ontology and Knowledge

- The Knowledge-level: a level of description of the knowledge of an agent that is independent of internal format.
  - An agent “knows” if it acts like it does.
  - A software agent “acts” by telling and asking.

- An agent commits (conforms) to an ontology if it “acts” consistently with the definitions
  - Ontological Commitments are agreements to use the vocabulary in a coherent and consistent manner.
  - Common ontology ≠ common knowledge.
What isn’t an ontology?

- a database or program
  - because they share internal formats
- a conceptualization
  - because it isn’t a specification - it’s a vision
- a table of contents
  - but wait, isn’t a Taxonomy an Ontology?
  - only if it defines a set of concepts
Ontology as Language

Language = syntax + vocabulary

- Ontology.org - XML based industry standards for e-commerce data exchange
- Gene Ontology, EcoCyc, etc. - for encoding complex biological data
The role of formalism

- Formal specification helps communicate the definition of terms in reader- and context-independent ways.
- Formal language semantics allows some automated consistency checks.
- Formal axiomatization is never sufficient.
  - It always comes down to the primitives!
Example Ontologies: Very Formal

- **EngMath** - basis for mathematical modeling of physical systems
  - *physical quantities, units, dimensions*

- **Frame Ontology** - unifying theory for frame-based representation systems
  - *classes, relations, slots*

- **Configuration Design** - for representing a design task
  - *components, subparts, attributes, constraints*
Example Ontologies: Semiformal

- **Biblio Ontology** - for unifying data about publications
  - *author, document, publication date*

- **CommerceOne’s Catalogs** - for describing products and integrating buyer services

- **Extricity’s Process Ontology** - for integrating processes across businesses
Example Ontologies: Informal

- **HTML Ontology** - for linking documents
  - *URLs and anchor tags*

- **Intraspect’s Context Ontology** - for capturing and sharing information in its context of use by knowledge workers
  - *parent/child, document, message, comment*
Representing Context of Use

- Marketing
  - Competitive Intelligence
    - Discussion about competition
      - Message about competitor
        - Competitor Web Page
          - Marketing comment
          - Engineering comment
  - Engineering
    - Research
      - Project X
    - Technology evaluation
The Intraspect Ontology

- Hierarchy with typed nodes
  - *allow multiple parents, no inheritance*
- Implicit metadata (author, date, filetype)
- Explicit metadata
  - titles and descriptions
  - user-defined types and attributes
- Conversational relations
  - next-in-thread/in-reply-to
  - context-sensitive annotation
Ontology as Content

Sometimes the ontology is a theory.

- VerticalNet, CommerceOne - catalog entries as the basis for netmarkets
- Yahoo ontology as real estate
- Library taxonomies - such as NLM initiatives for medical literature
- VT Ontologies - ontologies encoding sets of design constraints for elevators were used as inputs to knowledge services.
Ontology as Contract

Purposes of Ontologies
- data exchange
- unification/translation
- calling knowledge services
- representing theories
- human communication

Parties to the contract
- programmers
- data admins
- programmers, netbots
- scientists
- collaborators
Ontologies as Designed Artifacts
The Design Perspective

- Ontologies are designed to meet functional objectives
  - data exchange, unification, representation, communication …
- Representational choices are design decisions
- Design methodologies include validation, optimization against design criteria
General Design Criteria for Ontological Engineering

- **Clarity** - context-independent, unambiguous, precise definitions
- **Coherence** - internally consistent
- **Extendibility** - anticipate the uses of the vocabulary, allow monotonic extension
- **Minimal Encoding Bias** - avoid representational choice for benefit of implementation
- **Minimal Ontological Commitment** - define only necessary terms, omit domain theory
Wrap up

- Ontologies are what they do: artifacts to help people and their programs communicate, coordinate, collaborate.

- Ontologies are essential elements in the technological infrastructure of the Knowledge Age.