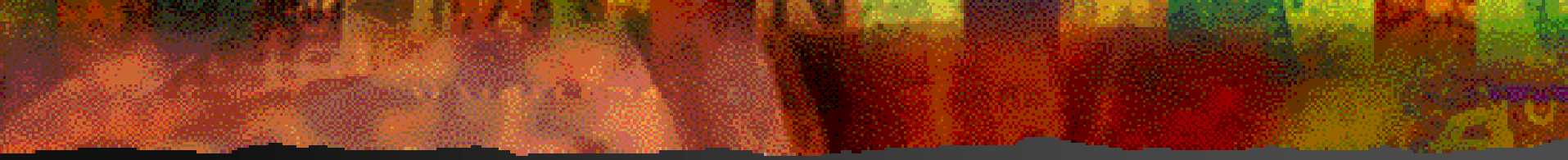


It Is What It Does: The Pragmatics of Ontology for Knowledge Sharing

Tom Gruber
Founder and CTO, Intraspect Software
Formerly at Stanford University
tomgruber.org

What is this talk about?

- What are ontologies?
  - *Theoretical perspective*
- What are they *for*?
  - *Pragmatic perspective*
- How do we build them?
  - *Design perspective*

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 language, the ontology defines a representational foundation.

Ontology, Knowledge, and Commitment

- 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 and Language

Language = syntax + vocabulary

- One can use the ontology as a representation language
  - *Penman ontology for natural language processing*
- *ML industry agreements

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

Formal => (partially) Computable Semantics

- **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

Semiformal => useful computations on formal part

- **Reference Dictionaries and Thesauri** - domain terms and untyped relations among them
- **Ontology.org** - XML based industry standards for e-commerce data exchange
  - *product, price, …*
- **CIDOC CRM** - conceptual reference model for cultural heritage data
  - *place, time span, appellation, right*

Example Ontologies: Informal

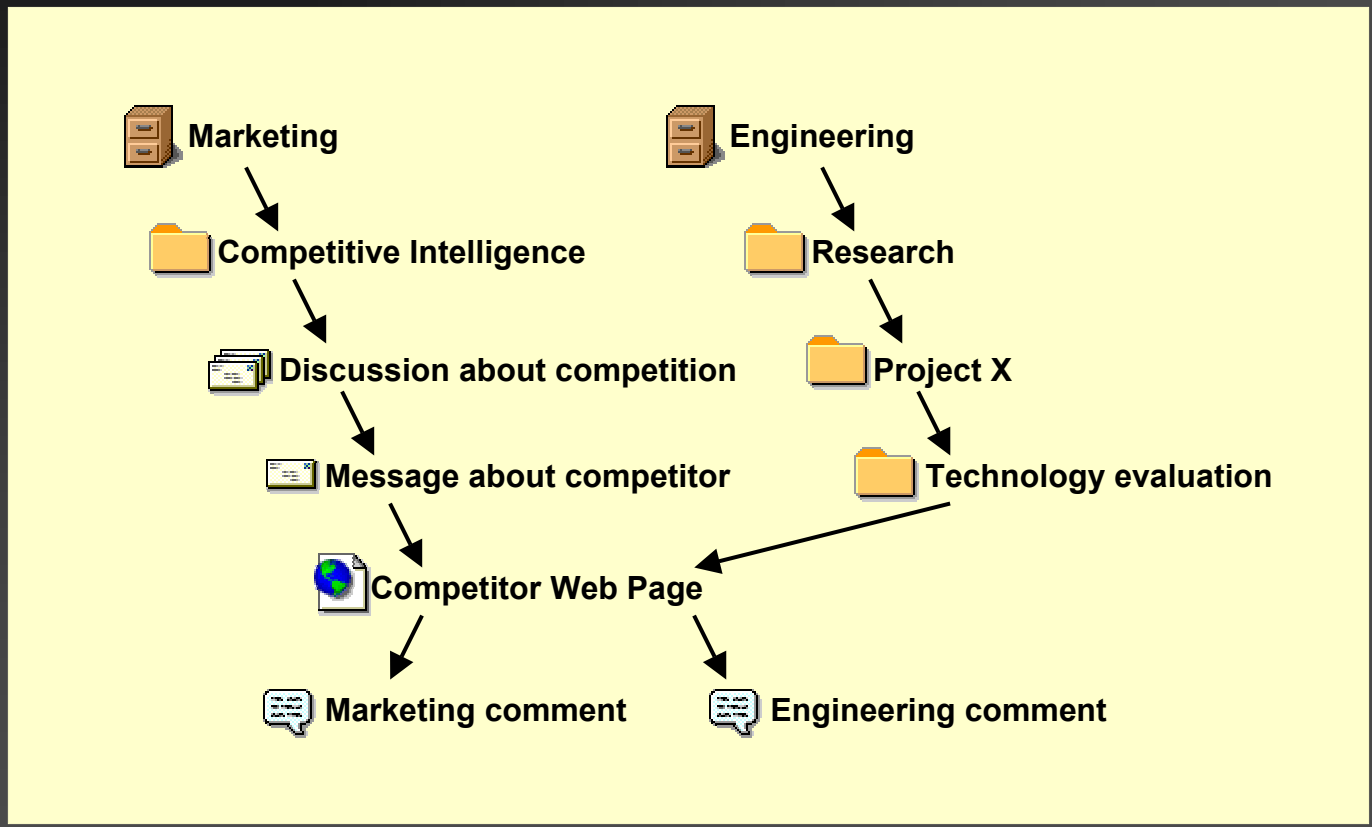
Informal => human interpretation aided by computation

- **(Non-semantic) Web Ontology** - for identifying and linking information objects
  - *Thing-with-URI, Link*
- **Intraspect's Context Ontology** - for capturing and sharing information in its context of use by knowledge workers
  - *parent/child, document, message, comment*

The Intraspect Ontology

- Hierarchy with typed nodes
  - *allow multiple parents, no inheritance*
- Implicit metadata (contributor, date, file type)
- Explicit metadata
  - *titles and descriptions*
  - *user-defined types and attributes (“deliverable”)*
- Conversational relations
  - *next-in-thread/in-reply-to (inferred from email)*
  - *context-sensitive annotation*

Representing the Context of Use



Why? Knowledge is created in context; information in context can be reused.

Ontology as Content

Sometimes the ontology *is also* a KB.

- Yahoo ontology as real estate
- VerticalNet, CommerceOne - catalog entries as the basis for netmarkets
- library taxonomies - such as NLM initiatives for medical literature (UMLS)

What are they for?

A Pragmatic Perspective

- Ontologies are not about truth or beauty.
- They are agreements, made in a social context, to accomplish some objectives.
- It’s important to understand those objectives, and be guided by them.

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
- as a reference to guide new formalizations
- to facilitate communication among people

Ontology as Contract

Purposes of Ontologies

- data exchange
- Unification and translation
- calling knowledge services
- representing theories
- human communication

Parties to the Contract

- programmers
- library scientists, database mediators
- programmers, netbots
- scientists, AI programs
- 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.
- We should design and build them ….

- for humans!