The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases*

Thomas R. Gruber
Knowledge Systems Laboratory
Stanford University
701 Welch Road, Building C
Palo Alto, CA 94304
gruber@sumex-aim.stanford.edu

January 31,1991

Abstract

Although AI research and commercial system development depend on bodies of formally represented knowledge that are expensive and difficult to construct, current knowledge base design does not support the accumulation or reuse of such knowledge. This paper presents a strategy for building libraries of sharable, reusable knowledge in which common ontologies play a central role as a knowledge coupling construct. Ontologies are defined as coherent sets of representational terms, together with textual and formal definitions, that embody a set of representational design choices. Problems in the design of sharable ontologies are identified.

Research and development in AI is impeded by an inability to share and reuse knowledge bases. The expense of building serious knowledge bases and the lack of means to exchange them with colleagues makes it difficult to generate, evaluate, and build on research results which depend on "domain theories" or "background knowledge." Similarly, commercial systems development is hampered by the lack of interoperability among tools based on different knowledge representations, and the lack of mechanisms for accumulating libraries of reusable knowledge.

Today's knowledge systems are isolated monoliths characterized by *high internal coupling* (e.g., among ground domain facts, procedures, terminology, axioms, and idiosyncratic ways of modelling the world) and a *lack of external coupling interfaces* that would enable the developer to reuse software tools and knowledge bases as modular components. The only practical way to "share" or "reuse" an existing knowledge base is to adopt the entire representation and programming environment of the existing KB.

*To appear in: Allen, J. A., Fikes, R., and Sandewall, E. (Eds.) *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference.* San Mateo, CA: Morgan Kaufmann, 1991.

What can be done to enable the accumulation, sharing, and reuse of knowledge bases? A proven software engineering approach is to *decompose* the monolithic system into reusable building blocks at modular boundaries. Three important decomposition techniques are already found in the AI methodology for building software systems:

- Separate knowledge from programs with a declarative knowledge representation language.
- Identify general classes and relations underlying application-specific facts, and organize knowledge to enable inheritance from these constructs.
- Characterize general problem solving tasks (e.g., classification) and classes of inference (e.g., subsumption), and design corresponding methods and algorithms.

Although they help, these techniques are insufficient to support sharability because the problem remains underconstrained. There are many ways to formalize declarative knowledge, organize class and relation hierarchies, and characterize tasks and inferences. To achieve sharable, reusable knowledge bases we also need to:

- Specify a **canonical form** for declarative knowledge: a representation language with a standard syntax and semantics (for operators such as $\wedge, \vee, \neg, \Rightarrow, \exists$). A proposal for such a language is described elsewhere [Genesereth, 1991].
- Define **common ontologies**: vocabularies of representational *terms*—classes, relations, functions, object constants—with agreed-upon *definitions*, in the form of human readable text and machine-enforceable, declarative constraints on their well-formed use. Definitions may include restrictions on domains and ranges, placement in subsumption hierarchies, class-wide facts inherited to instances, and other axioms.

The purpose of this strategy is to enable the "literary" publication and exchange of formally-represented knowledge.[1] The canonical form provides the linguistic foundation, drawing a level boundary that factors out differences in *notation* among knowledge bases. Ontologies capture reusable intellectual *content* of a representation effort—the choices about classes and relations that are potentially relevant for describing a domain or performing a task. Each ontology embodies a set of ontological commitments in a form that enables one to build knowledge bases and tools based on those same commitments. Given a common language and vocabulary, one can build knowledge bases that instantiate and specialize the shared classes and relations. The instantiations and specializations of the ontologies carry application-specific information. Thus, the role of ontologies is to specify a *modular coupling* among bodies of knowledge and the tools that operate on them, serving as knowledge-level protocols for input, output, and communication.

The aim is to build *libraries* of shared, reusable knowledge. If the specification of a standard declarative language is like a grammar of English, ontologies are reference

[1] Recent activities along these lines are described in [Gruber, 1990] and [Neches, 1991].

works akin to dictionaries. Libraries could contain "off-the-shelf" knowledge-based tools that perform well-defined tasks such as varieties of simulation, diagnosis, etc. Ontologies specify the terms by which a tool user writes the "domain knowledge" for the tool, such as the equation models that drive a simulation or the components and failure mode descriptions used by the diagnostic engine. A knowledge library could also contain reusable fragments of domain knowledge, such as component models (e.g., of transistors, gears, valves) that can be composed to produce device models (e.g., of amplifiers, servo mechanisms, and hydraulic systems). Ontologies define various ways of modelling electrical, mechanical, and fluid flow mechanisms that make such reusable component libraries possible.

Like reusable software, knowledge bases have to be designed for sharability, organized by ontologies designed to support reuse. Mechanical translation of existing knowledge bases is no panacea; the reusable content of a knowledge base must be teased out and formulated to minimize hidden assumptions. The defined terms in ontologies identify and deliver the product of the representation effort, which is reused when the terms are instantiated in new applications. Note that the leverage does not come from reusing entire knowledge bases, filled with millions of ground facts about a broad domain of discourse (c.f. [Lenat and Guha, 1990]). Knowledge-based systems will always require application-specific knowledge, and no effort could hope to anticipate all possible content. The leverage comes from applying the ontologies that embody the representational choices underlying such knowledge bases in the design and construction of similar knowledge bases for different institutions and domains. Making such ontologies explicit and public can create a market for tools designed to operate on domain knowledge represented using a well-defined, agreed-upon vocabulary.

What are the technical problems in designing a common ontology? Space permits only an incomplete enumeration.

- What information about terms is most critical for supporting sharability? The names? Textual definitions? Type, arity, and argument restrictions? Arbitrary axioms?

- How can we achieve group consensus on "what to represent" when participating researchers have commitments to different tasks, representation tools, and domains? How can one describe assumptions and purposes of a particular ontology?

- How can one capture and use design rationale for representation (ontology) design?

- What kinds of automated assistance can support the development of consistent sets of terms in a collaborative setting?

- What mechanisms can be used to verify compatibility with an ontology?

- How does one write correspondence theories that relate different ontologies (e.g., multiple ways of modelling device behavior)?

To investigate these and other questions we conducted a pilot study in the collaborative development of ontologies. In the Summer Ontology Project [Gruber, in preparation], computer scientists and engineers from several groups at Stanford and Bay Area research labs met to develop common ontologies for describing a class of electromechanical systems. We focused on four existing "motion control" devices, attempting to model their physical structure and dynamic behavior with several languages and approaches. From an analysis of these models and many hours of group discussion we have begun to define a set of representational terms. To accommodate the differing representation styles of the participants, we developed a translation mechanism called Ontolingua by which term definitions and axioms in the standard interchange representation can be automatically transformed into the corresponding forms for implemented frame systems and predicate calculus languages. The primary outcome to date is an ontology for lumped-parameter behavior models of physical devices that encompasses several modelling languages used in AI and Engineering. As a side product we defined an ontology for class/instance/slot representations: a portable vocabulary that unifies notions of classes and instances, relations, slots, and metaslots. When complete these ontologies and others under development will be distributed for evaluation and experimental use.

Acknowledgements

This work was supported by Grant NCC2-537, NASA Grant NAG 2-581 (under ARPA Order 6822), IBM Agreement No. 14780042, and NIH Grant No. LM05208. Thanks to the many participants of the summer ontology project and the DARPA KRS effort, especially Mike Genesereth, Bill Mark, and Marty Tenenbaum. Space limitations preclude a decent accounting of the relevant literature; a longer version of this paper will acknowledge the sources of many converging ideas.

References

Genesereth, M. R. (1990). DSL Reference Manual. Stanford University, Computer Science, Report Logic-90-3.

Genesereth, M. R. (1991). Knowledge Interchange Format, Version 2.1 Reference Manual. Stanford University, Computer Science, Report Logic-90-5.

Gruber, T. R. (1990). The Development of Large, Shared Knowledge-Bases: Collaborative Activities at Stanford. Stanford Knowledge Systems Laboratory, Technical Report KSL 90-62.

Gruber, T. R. (in preparation). An Experiment in the Collaborative Development of Shared Ontology. Stanford Knowledge Systems Laboratory.

Lenat, D. B. and Guha, R. V. (1990). *Building Large Knowledge-based Systems.* Addison-Wesley.

Neches, R. (1991). Knowledge Representation Standards Effort. Forthcoming in the *AI Magazine*.