SPECIFICATION OVERLAYS: AN APPLICATION OF PARTITIONED SEMANTIC NETWORKS

Robert L. Young
Department of Computer Sciences
University of Texas
Austin, Texas 78712

Introduction

Specification Overlays are a technique for representing knowledge hierarchically, where the "nodes" are structured concepts. The scheme is described in terms of partitioned semantic nets suggested by Hendrix [75]. The idea of multiple interpretive contexts is also found in a very general form in several A.I. languages (see Bobrow and Raphael [74]).

Specification .Overlays

Knowledge is organized in a number of shallow hierarchies, each representing a complex concept. The root of each of these trees is a kernel concept and each lower node designates some specification to a more particular concept. The kernel concept represents the canonical or fundamental form of the concept. A specification can be visualized as the application of an overlay to the kernel, which adds and modifies information, with the result seen as a single, unified concept. A reference to (or into) a concept is made to the kernel, with an explicit indication of the appropriate sequence of specifications. For example, an instance of SHOOT would be a simple TOK arc to SHOOT. But an instance of SHOOT-OLD-CANNON would be represented by a TOK / (SHOOT-OLD-CANNON, SHOOT-CANNON, SHOOT-WITH-GUNPOWDER) arc to SHOOT. The convention followed is that the specification path, from most specific to least specific, is given following the slash of the TOK arc pointing to the kernel concept.

An Implementation

A simple, efficient implementation has been based on GRUNT2 (Smith and Young [77]), a set of Lisp functions which provide efficient representation and manipulation of partitioned semantic nets. A partition in GRUNT2 is called an arcset, and each arc in the net is in one or more arcsets. All accessing functions are sensitive to the list of currently active arcsets, which is called the context. From a given node, the only arcs accessible are those in some arcset in the current context. All computed lists of arcs or nodes are ordered according to the context.

A kernel concept is represented essentially as a single (default) arcset. Each specification of the kernel concept is defined by a separate specification arcsqt which contains the relevant modifications. The simplest method of effecting a modification is through the use of a replacement arc in a "more visible" (nearer the front of the context) arcset. For example, the Instrument of SHOOT has a TOK arc to MACHINE, but the SHOOT-WITH-GUNPOWDER arcset has a TOK arc to GUN. For an instance of a concept, the specification path arcsets and the kernel arcset form the context used in accessing that concept. Thus, if SHOOT-WITH-GUNPOWDER is referenced, the TOK arc to GUN is visible, and being "more visible", is used. Modification of the kernel concept can also be achieved in a number of other ways, such as AND, OR and NOT arcs in specification arcsets.

Advantages

One advantage of the described technique is the ability to treat a particular concept as a single, unified concept. This eliminates the explicit traversing of SUP arcs within an algorithm. Instead, the algorithm is expressed in terms of an exploration of the complete concept. Controlled accessibility accomplishes this without construction of the concept whenever a reference is made. This composite already exists in the representation. A second advantage afforded by Specification Overlays is a clear delineation of the central meaning of concepts, similar to the use of semantic primitives. The notion underlying the use of kernel concepts is that each complex concept may be treated as a (relative) primitive, with a variety of possible elaborations. A third advantage is the simplicity with which certain pattern matching may be performed. Any pattern requiring a concept K with specification path (X) will be matched by any concept K with specification path (YX), and this is immediately recognizable because of the reference representation. The usefulness of such matching is due to the use of the kernel concept to capture the central meaning of concept. A final advantage is that it is possible to simply describe some aspects of forgetting. Generalization and loss of specific detail in the representation of an instance of some particular concept is modelled by dropping specifications from the specification path. For example, a particular train trip would be represented as an instance of GO with specification path (RAIL, LAND-TRAYEL). Dropping RAIL would result in the representation of a land trip, and additionally dropping LAND-TRAVEL would leave a totally unspecified GO. Further generalization would be impossible. Generalization also creates the possibility of sub-events becoming inaccessible, which would seem to be the type of loss that should occur in forgetting by generalization.

Current investigations in text representation using Specification Overlays should test and demonstrate these capabilities.

References

Bobrow, D. and Raphael, B. 1974. "New Programming Languages for Artificial Intelligence Research". Computing Surveys, 6(3), 153 - 174.

Hendrix, G. 1975. "Expanding the Utility of Semantic Networks Through Partitioning." Advancepapers*Ql the Fourth* International Joint. Conference *on* Artificial intelligence.

Smith, M. and Young, R. 1977.The Annotated Grunt. unpublished reference manual.