THE USE OF A DOMAIN MODEL IN UNDERSTANDING INFORMAL PROCESS DESCRIPTIONS[1]

Neil Goldman, Robert Balzer, and David Wile
USC/Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA. 90291

INTRODUCTION

The SAFE project [Balzer 77] at ISI is constructing a computer model which exhibits certain aspects of process understanding. SAFE accepts a written, pseudo-English process description and produces a formal specification of the described process. The resulting specification is encoded in a language with formal syntax and semantics. The process is specified in operational terms in this language. In this sense the formal specification resembles a computer program, although it does not incorporate commitments to data representation or considerations of input/output, and thus is not suitable as a software implementation. This lack of commitment makes the formal specification a useful object for reasoning about the behavior of the process, however, since implementation details necessitated by programming languages, efficiency considerations, and the architecture of digital computers are not an inherent part of the specification.

MODELING A PROCESS

Underlying our factoring and description of process knowledge is a simple view of what constitutes a process. Although execution of a process may involve manipulation of concrete objects and specialized information representations demanded by the execution environment, understanding the formal structure of a process involves an abstraction from the physical process. At that abstract level, a process is a controlled application of ACTIONs to OBJECTs. The effect of applying an ACTION to OBJECTs may be to directly invoke further ACTIONS, to create new OBJECTs, and to create or destroy ASSOCIATIONS between OBJECTs. The environment in which the process operates consists of a relational data base [Chamberlin 76] of these associations.

MODELING A DOMAIN

A domain model consists of information about types of objects in the domain, the classes of associations *(relations)* which may be formed between those objects, constraints on the set of associations which may co-occur, inference rules which permit the derivation of new associations from old, the actions which may be performed, along with their operand types and pre- and postconditions, particular object instances, and particular associations between those instances. This information may be characterized as:

- time-independent — Whereas any process creates and destroys associations (information) as it operates, the domain structure information remains static.
- constraining rather than determining -- Each piece of a domain model serves to constrain the universe of processes which can be built within the domain. The domain structure does not determine a particular process, however, but only an infinite class of processes.
- needed for the non-performative aspects of process understanding rather than for actual process execution -- A process within the domain must conform to the constraints imposed by the domain, but has no need to access the descriptive information in its operation.

USING THE DOMAIN MODEL

Domain models have been used in AI planning systems [Tikes 76] and have been suggested as one approach to providing question-answering facilities for data base management systems [Sowa 76]. The former relies mainly on the characterization of the domain's actions, while the latter depends on the characterization of relational connections in the domain and functionality constraints. Understanding descriptions of processes, particularly when those descriptions are provided in an informal language, appears to require a stronger model of the process domain than those utilized for the planning and question-answering tasks. Such descriptions, unlike conventional computer programs, do not generally contain sufficient explicit declarative and procedural information to ensure unique interpretations of individual phrases and statements.

- Reference to objects is not by (variable) name, but by description.
- Operands may be omitted where the process describer feels that the identity of the "proper" operand is "sufficiently obvious".
- The proper sequencing of actions and tests may not be explicitly specified, even when that sequencing is not arbitrary.

SAFE understands such descriptions by using a model of the process domain, either pre-existing or acquired from the description, to select and combine plausible interpretations of informal language constructs.

First, the domain model is used to suggest local resolutions of informalities -- e.g., fill in missing operands with objects of appropriate type; for unnamed relationships between described objects select relations known to relate objects of the types described. A consistent set of these resolutions is then sought by executing the hypothesized program on symbolic data. The pre- and postconditions and constraints in the domain model help force rejection of choices which may have appeared satisfactory in a static context.

REMARKS

The construction of a formal domain model for any domain of sufficient size to be interesting would be an arduous and error-prone task, for which no realistic completeness test can be given. Fortunately, process descriptions (both formal and informal) reflect the underlying domain structure. It is therefore possible to construct/augment the domain model from a process description itself [Goldman 77].

REFERENCES

[1] Balzer, R., Goldman, N., and Wile, D. *Informality in. program specifications.* USC/Information Sciences Institute, ISI/RR-77-59, April 1977.

[2] Chamberlin, D. Relational data base management systems. *Computing Surveys,* 8, I, 1976, pp. 43-66.

[3] Fikes, R. *Knowledge representation in automatic planning systems.* Stanford Research Institute, T.N. 119, 1976.

[4] Goldman, N., Balzer, R., and Wile, D. The inference of domain structure from informal process descriptions. Proceedings of the Workshop on Pattern Directed Inference Systems, *SICART Newsletter 63,* 1977.

[5] Sowa, J. Conceptual graphs for a data base interface. *IBM Journal of Res. and Dev.,* 20, 4, July 1976.

[1] This research was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. DAHC15 72 C 0308.