KNOWLEDGE BASE DEVELOPMENT
IN A STANDARD FRAMEWORK

GIUSEPPE ATTARDI and MARIA SIMI
DELPHI S.p.A.
Via della Vetraia, 11
I-55049 Viareggio LU
Italy

Summary

In the ESPRIT Project P440 (Message Passing Architecture and Description Systems) several aspects of knowledge representation have been explored: formalisms, languages, techniques. The most significant techniques developed within the project are: taxonomic reasoning, meta level strategies and concurrent object oriented programming. These techniques are embedded in the systems developed in the project: Omega, KRS and DELPHI Common LISP. As a generalization of our experience in building knowledge based systems, we present a framework which provides a range of tools for building knowledge base applications. The framework is conceived so that industrial standards are selected where appropriate and areas where standard are missing are highlighted.

1. Introduction

The ESPRIT Project P440 (Message Passing Architecture and Description Systems) aims at exploring techniques to push the limits of knowledge base systems in terms of performance and size.

We worked on the assumption that complex knowledge bases could only be managed if suitable structuring mechanism are provided so that the search and deductions could be focused and properly directed. In addition, concurrency would be used to speed up the search both through the various paths in the knowledge base and by applying alternative strategies.

The project has gone through the following phases:

1. design of suitable formalisms for knowledge representation
2. development of techniques for reasoning on the knowledge base
3. verification of the techniques
4. generalization of the results

The formalisms developed are Omega, a logical calculus of descriptions, and KRS, a programming language for knowledge representation.

The techniques that have been developed are:

1. This work has been partially supported by ESPRIT Project P440 (Message Passing Architectures and Description Systems) and is a contribution to Cost13 n.21 (Advanced Issues in Knowledge Representation).
1. taxonomic reasoning, a form of deduction which explores a taxonomy of concepts
2. meta level strategies, which allow to tailor the reasoning strategies to each specific task
3. parallel constructs for spreading the deduction across the nodes of the taxonomy

The sperimentation phase has produced various software systems, which have been made available also outside of the project: the Omega Knowledge Base Development Environment, the KRS language and an implementation of Common LISP (DELPHI Common LISP) which includes a Multithread facility for concurrency (3) and the Common LISP Object System for object-oriented programming.

The generalization of the experience has led to the design of a framework for building knowledge base applications which is based on industrial standards, and which incorporates as "generic" tools the essential building blocks that were used in the construction of Omega and which can be helpful in a variety of knowledge base applications.

In the following sections we will put our approach into the perspective of knowledge representation research, we will sketch the techniques developed in the project and finally we will illustrate the structure of the knowledge base development framework which constitutes a generalization of our experiences.

2. Perspective on Knowledge Representation

Historically the research in knowledge representation started investigating the problems of formulation, i.e. how to translate into a precise formalism the expressions of natural language, the same problems that philosophers and linguists had been debating for long time: reference, co-reference, definite or indefinite descriptions, referential transparency and so on.

The issues addressed were of the kind: "How one would represent statements like 'the present king of France is bold' or 'Kepler did not know that the number of planets is 9'?".

Experiments with deductive systems that were being conducted at the same time, demonstrated however that even for problems whose formulation in predicate logic was straightforward, the computational demands for even the simplest deductions were overwhelming.

So while research is still continuing in problems of formalization, and new issues like default reasoning, nonmonotonic reasoning, reasoning about beliefs, about time, etc. have been added to the list of unsolved problems, an alternative line of investigation has emerged.

This research addresses the problem of developing formalisms with a proper balance between the overall expressive power and the computational complexity of the basic constructs. A simpler formalism is expected to be more amenable to algorithmic treatment, for instance as the basis for an automated deduction system.

Two approaches have been followed, a "programming language" approach and a "logic formalism" approach.

In the first approach, the basic formalisms were programming languages which were designed or extended to provide constructs useful to represent knowledge (6),(19). Typically new constructs were introduced to define objects and classes of objects, usually related by some kind of inheritance hierarchy. These capabilities were useful to model the individual objects of the domain of discourse, and programs could be written to manipulate those objects. Beyond these basic descriptive capabilities, the bulk of the semantic knowledge on the domain, would be expressed by procedures associated to the objects, or sometimes attached to the slots of some object. For instance in LOOPS one could