Statement of Interest

Bob Dionne, Sitaram Lanka, Eric Mays, Tony Weida
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
1989

The increasing size and scope of knowledge based systems places great demands on the tools employed to develop and deploy these systems. A useful analogy might be drawn between the data processing tools of the 1960’s and the knowledge processing tools we have been using in the 1980’s. Data processing in the 1960’s largely consisted of batch processed, application specific data structures. Communication between applications was obtained by passing files. In the 1970’s data base management systems were introduced to provide logical views of data, shared across many applications. Data base management systems provided applications independence from the lower level access mechanisms used to store data. Knowledge processing in the 1980’s is characterized by application specific knowledge representations (or even knowledge representation languages) operating in a stand-alone mode. Future knowledge processing systems will see an ever increasing demand for knowledge reuse in the development cycle and knowledge sharing in the deployment phase.

For the past few years we have been developing a knowledge based system, FAME[4, 5], intended to assist IBM marketing representatives in the financial marketing of large scale computing systems. We have built a semantic network based knowledge representation facility, K-Rep, which incorporates an enforced semantics and classification, as in KL-One[1, 2]. This style of knowledge representation encourages a disciplined method of knowledge engineering, forcing more careful representation decisions than would otherwise be made in a system without enforced semantics. We have had modest success at knowledge reuse, in that we have built a financial planning problem
solver, a capacity planning problem solver, a user interaction facility, and a domain specific browser that all share a common knowledge base. We believe that the enforced semantics of the representation facility has greatly contributed to this achievement.

Today, the initial FAME knowledge base consists of roughly 1500 concepts with an average 34 local roles per concept. During a typical problem solving session hundreds of additional concepts are added dynamically to the knowledge base. However, we have covered only large mainframe computers in this knowledge base. Support for an entire product line, including configuration and software, would require on the order of 100,000 concepts. A knowledge base of that magnitude is not achievable in today’s representation systems. We have recently begun to explore some of the issues involved in the development of a knowledge base management system (KBMS) which would support a knowledge base of that size. Our current design point is oriented towards a KBMS which incorporates version oriented concurrency control mechanisms (as in CAD data bases), along with an incremental KB storage manager. The central part of a KBMS is the consistency checking mechanisms, and facilities for the arbitration of conflicting updates made by knowledge engineers. The development of large knowledge bases requires the coordinated effort of several knowledge engineers. Consistency must be managed throughout the lengthy transaction time required for a knowledge engineer to update a knowledge base. The knowledge based systems area offers a unique perspective on this problem, since a well defined semantics for representation languages strengthens the notion of consistency. That is, the knowledge based setting strives to share knowledge rather than just share data, thus allowing the KBMS to appeal to inference in addition to locking.

The important and novel issues relating to the design and implementation of a KBMS relate primarily to the performance aspects of accessing a large KB from secondary storage and incrementally maintaining consistency and re-classification. In the area of secondary storage we are concerned with minimizing the amount of storage required to create a new KB version, efficiently accessing any given version of the KB, and efficient updates to the knowledge base. Furthermore, it is necessary to store and access the KB in such a way that repetitive access to blocks on secondary storage is eliminated. The consistency of the KB must be maintained. An algorithm for consistency checking which requires visiting the entire KB is unacceptable. Our approach is an incremental algorithm which on the average visits a small
subset of the KB. Similarly, classification is a potentially expensive inference whose complexity must be managed.

Additionally, knowledge based techniques can be employed by a system attempting to facilitate the knowledge engineer’s collaborative work process. This presents itself in the KBMS setting at the time of merge. If a consistent merge of knowledge base version cannot be obtained a heuristic may be applied to attempt to isolate if some small number of updates could be rolled back to allow for a consistent merge. A large knowledge based system requires the collaborative work of knowledge engineers. For example, the current FAME knowledge base covers only a small part of the product line with the need to update financial parameters on a monthly basis and incorporate new tax strategies yearly, perhaps across an international environment. The amount of expertise required spans many individuals and organizations. Knowledge based support of the collaborative process is essential.

References


