Using a Knowledge Representation Language to capture both Knowledge and Routine Data

Anne E. James and **Richard A. Gatward** and **Steve Shipley**
School of Mathematical and Information Sciences
Coventry University, Priory Street, Coventry
Tel. 0203 838991, Fax: 0203 221608
E-mail: csx118@uk.ac.coventry

Database systems typically have a simple structure designed to facilitate the management of large amounts of abstracted, structured data with a high degree of uniformity. Knowledge representation languages on the other hand typically embody much greater richness with the aim of reflecting information that is not quite as abstracted or structured and which has a far lesser degree of uniformity. We contend that although the simplicity of structure of current database systems is useful for coping with many routine tasks, systems of the future should be able to combine the management and processing of routine data with the reflection and utilisation of knowledge. This should result in more sophisticated systems that are more user-supportive and less prone to human error. We are currently involved in a project which aims to combine knowledge and database techniques for modelling engineering applications.

The approach we are taking involves the use of a knowledge representation language, CPL (Conceptual prototyping Language, see [1]) to capture both knowledge and routine data. CPL is based on linguistic theory (Functional Grammar, FG, see [3]) and uses the semantic basis of predicate calculus. The motivation behind the development of CPL was to produce a knowledge-based modelling language that had the power to express any kind of knowledge that one might want to incorporate into a system. In particular CPL includes the implementations of logics to allow for the specification of vague knowledge, knowledge about events and obligations and knowledge about temporal aspects [2].

One of the key ideas of FG used in the development of CPL is that of the semantic function. This is used in the context of CPL to specify roles defined by the use of certain verbs (called relations in CPL) in the application domain. Capel and Wistra [1] give a list of semantic functions used in their interpretation of the language. These include both those defined in the theoretical specification of Functional Grammar, along with those that they have added for the purpose of the modelling language. The kind of semantic function that can be applied to a particular slot in the predicate depends upon the nature of the slot. Predicates that have been extended with satellites [3] can have a different set of semantic functions applied to the satellite position from those that can be applied to a basic predicate. In the theory of functional grammar the parts of the expression represented by satellites roughly corresponds to prepositional clauses and similar appendages that are added to the basic utterance to give additional information about a stae of affairs to that which is required as a minimum by the use of the main verb in the expression. CPL has some special relations such as 'is-a' and 'has' where the semantic functions are predefined. In other cases the user will select appropriate semantic functions for a relation.

CPL is very rich and mirrors natural language structure. Since we can express most information in natural language it follows that CPL, as a formalised version thereof, can be used to express most knowledge. There is a problem however, in that operational information that typically needs to be recorded is more conveniently recorded using simpler frameworks. Therefore we propose to use CPL as defined for knowledge representation and an extension thereof for handling routine, uniform data sets.

The proposed extension to CPL introduces an explicit meta-level for defining routine data according to the object-oriented paradigm. A new statement type METAFACTUAL will allow for the definition of uniform data sets. This statement type will be used to define object classes, their operations and object class sub-type and containment hierarchies. Special relations with predefined semantics such as 'is-object-class', 'is-operation', 'has-operations', 'has-objects', 'is-sub-type' and 'is-instance' will be introduced for this purpose. An implementation of extended CPL would then need to include the operational semantics of the object-oriented approach as well as that of functional grammar and predicate calculus.

The above gives an overview of the type of approach we are taking to integrating knowledge and database techniques. We feel the approach is novel in that most other work we have seen adds knowledge constructs to database model formalisms wheras we have taking the opposite approach of extending a knowledge representation formalism to capture data model concepts. The work is at an early stage and planned future work will involve further prototyping of the ideas, defining the necessary CPL extensions more rigourously, examing the underlying semantics of a combined formalism and developing suitable user interfaces.

References

[1] Capel C. and Wistra D. LIKE, MSc Thesis, V U Amsterdam, 1987.

[2] Dignum F. and Van de Riet R. P. Knowledge Base Modelling Based on Linguistics and founded in Logic. *Data and Knowledge Engineering*, 7, pp 1-34, 1991.

[3] Dik S. C. Functional Grammar, North Holland, 1980.