DISCUSSION

Authors: Sargent, Subrahmanian, Downs, Greene, Rishel

Yoshio Monma: We are developing a system using a similar approach, though our target is more specific and we have experienced strange side effects of inheritance. "Do not overuse the inheritance" is our conclusion. Do you agree?

Philip Sargent (author's response): We are using inheritance only for explanations, definitions and other descriptions of technical terms and concepts. It is true that inheritance algebras can be very complex, but we are careful to use inheritance as simply as possible.

Ranganath Shastri: Most of the time, the database developers overlook the fact that design engineers do not need more data - what they ask for is one piece of specific data which provides a sense of confidence/reliability indicator about the quality of data. It is heartening to hear that you have given some thought to this. In light of your comment that conceptual modelling approach is a good base for database developers but not end-users, would you care to elaborate a little on your experience on feedback from end-users and design engineers? Further, how do you propose to address these issues?

Philip Sargent (author's response): I must admit that I am not personally aware of feedback from design engineers at Alcoa Technical Center. The project had not yet reached a phase where feedback was appropriate when I left Pittsburgh. I suspect that a separate user interface which makes more prominent use of aliases and related entities would be appropriate.

Norman Swindells: 1) It is not clear whether your conclusion referred to the application of the CODE software to this particular problem or to the software itself. Could you clarify this please? 2) At the CEC Joint Research Centre Petten we were planning to apply CODE to the problem of multi-lingual terminology relationships. Would you have an opinion on its effectiveness for this purpose?

Philip Sargent (author's response): 1) The conclusions apply, I think, to any conceptual modelling tool of the CODE-type. Management of inheritance relationships is complex and the complexity here is inherent in the real world we are trying to represent and so should be independent of any particular software tool. CODE is very flexible, so flexible that we used only perhaps one third of its facilities. With more familiarity we might have found a different subset to be more appropriate. Finding the most appropriate representations for materials concept taxonomies is a research issue and we have made only one beginning.

2) The multi-lingual problem is easily handled when there are exact translations of terms: the usual alias mechanism is appropriate. However, that is not really the answer: the real problem with translation is partially-synonymous terms. I must admit that this important issue is not handled well by our use of CODE in the current project, and that this will
have to be dealt with as the project at ATC moves towards a data directory accessible to Alcoa end-users. I strongly suspect that CODE contains appropriate facilities to manage partial and overlapping synonyms, but we have not used them yet.