

COMMON INTERFACE BETWEEN PROGRAMMING LANGUAGES  
AND DATA BASE MANAGEMENT SYSTEMS  
BY: JONAS RABIN AND CHENG W. CHENG

Adaptation of a common interface would allow application programs to be written which would have some degree of transferability from one data base manager to another. This would give more flexibility in switching either implementations of data base managers or even hardware vendors. This would also minimize programmer training, especially in those installations where multiple vendors run side by side with a common set of supporting personnel. More readable source code and possible reduction of the amount of source code would reduce the cost of development, modification and maintenance of applications.

The components involved in such an interface are: Schema, Subschemas, Data Manipulation Language (DML) and Program Communication Areas. One approach is to concentrate on defining a common DML with descriptions of data structures to be used by data administrators to provide the appropriate schema and subschemas. A pre-processor could be built which would accept, as input, host language statements and common-DML statements. The pre-processor would build program communication areas (working storage structures and linkages to the appropriate data base manager). The pre-processor would perform the mapping (1-1 or 1-n) between common-DML statements and DML statements of the appropriate data base manager.

STRUCTURED PROGRAMMING AND INTEGRATED  
DATA-BASE SYSTEMS DESIGN  
BY: JONAS RABIN AND CHENG W. CHENG

In the literature on structured programming a module is essentially defined as consisting of a number of functions which communicate with a common data structure. The modules are independent in the sense that the functions of one module do not produce side effects in the data structure of another module. An integrated data-base system is based on the concept of sharing of data among all the application programs. Structured programming is a tool to increase system reliability and maintainability. The problem is to find tools in aiding the development of reliable and maintainable systems where sharing of data is involved.

A module could be associated with each data item. But this could become practically unmanageable due to a large number of modules