Frank, A. U. "Databases Are Crucial Tools for Computer Aided Software Engineering." *First International Workshop on Computer-Aided Software Engineering* 2 (1987): 531-32.

First International Workshop on Computer-Aided Software Engineering

Advance Papers

Volume 2

Cambridge, Massachusetts
May 27-29, 1987

#131

CASE 87 Position Paper

Andrew U. Frank, University of Maine
Civil Engineering Department
103 Bboardman Hall
Orono, Me 04469

Databases are crucial tools for computer aided software engineering.

A software project consists of a large amount of structured expression in formal or informal languages, and software engineers, the users of the CASE tools need access to these using different access paths, different level of detail and concern. Traditionally, a database management system (DBMS) is a method to store large amounts of structured data with access along multiple access paths. Thus, a CASE system must include a DBMS. A DBMS, or at least DBMS technology, is necessary in a CASE system in order to provide:

-multiple access paths,
-flexible structuring of data allowing multiple forms of presentation,
-multiple users within a project, with read and update access at the same time,
-protection of the data from accidental loss (due e.g. to malfunctions in the hardware),
-security of the data from non-authorized access or alterations,
-distribution of a single logical database over several connected computer system, with control of replication, consistency etc. (distributed DBMS).

Unfortunately, the currently available DBMS do not provide all the services the CASE system needs:

-complex consistency constraints (e.g. the syntax rules of programming languages). The database must be able to cope temporalily with inconsistent states and detect them later for clean up,
-long transactions; considerable amount of work may be necessary to move the project from a consistent state to the next (e.g. a change in a low level module which requires adaptation in many places); the transaction concept in DBMS used for protection of data and control of

concurrency yields unacceptable results, as only completed transactions are preserved,
-management of history of changes; it seems desirable to store the successive changes applied to a project for example to be able to reconstruct previous states or to understand dependencies
-multiple levels of detail and different views of the same object (e.g. levels of increasing specification from high level specs to programs),
-changes may be applied tentatively and only after approval is incorporated in the overall version of the database; this requires more complex rules for visibility of data than presently employed
-changes may be visible only to the author, optionally visible to his group (for study and approval), visible to some other groups, etc.

These requirements and problems with present DBMS, which were primarily developed for commercial applications, are not new. Very similar demands are posed from CAD/CAM application for data storage. This should not surprise, as the design of a building or a machine is certainly somewhat similar to the construction of software. In the database world the topic is treated under the label of 'non-standard' or engineering databases. Unfortunately not all the problems are solved, but the consideration of the existing literature on engineering databases is recommended for builders of CASE systems.

The author, with a group of students, has started building a CASE system based on a database, which was previously designed for engineering applications, and we were surprised by the similarity and the complexity of the problems. If interest exist, we may report about the database schema used and other results.