

REPORT ON IFIP TC-2 CONFERENCE

"A Technical In-Depth Evaluation of the DDL"

Wepion, Belgium, Jan. 13-17, 1975

Robert W. Taylor
Computer and Information Science
University of Massachusetts, Amherst

In response to increasing interest in data management, Technical Committee 2 of IFIP has been sponsoring a series of conferences over the past year. The most recent of these was held in Wepion, near Namur, Belgium, and dealt primarily with the Schema Language defined in the CODASYL Data Description Language Journal of Development. The CODASYL Schema Language has been widely discussed in the U.S. (witness the debate at the ACM SIGMOD Conference, held last May in Ann Arbor), and the interest in Europe on this same topic is at least as great. This conference was somewhat unique, however--the focus of the entire conference was on the single topic of the CODASYL DDL, hence the topic was discussed in more detail than probably it ever has been outside of DDLC (Data Description Language Committee) itself.

Present at the conference were several DDLC members, as well as members of the DDLC's predecessor, the Data Base Task Group. In particular, Charles Bachman, "Tax" Metaxides, Christopher Earnest, Jim Lucking, Teresa Gerrasimenko, and H. Randall Johnson are or were members of DBTG, DDLC, or affiliated task groups. Also present were about 50 representatives of both universities and industry from most of the Common Market countries, as well as the U.S.

The main theme of the conference was established on the first day in a paper given by G. M. Nijssen of Control Data Corporation, who was general chairman of the conference. Nijssen proposed that the industry work toward a data description language (and associated data model) which was sufficient to support any of the commonly used views of data--hierarchical, relational, network--and powerful enough to accommodate other views,

e.g., binary relations. In addition, this common description or "conceptual schema," as it was called most of the week, should support both a "navigational" user and a "higher level" user. Mr. Nijssen argued that both types of user must be supported since:

- 1) A procedural language is sometimes "easier" (see below).
- 2) The higher level interface is desirable in certain situations and feasible in all data models.

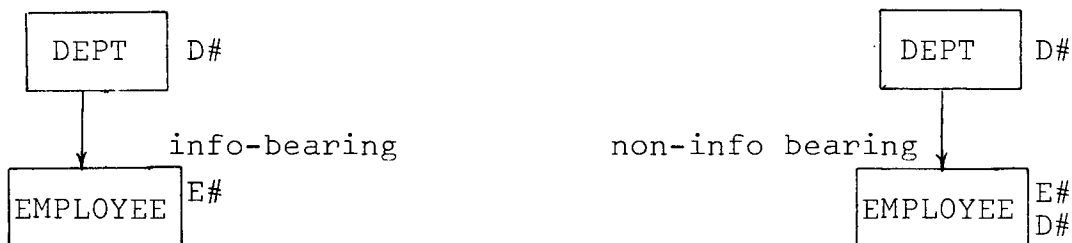
Mr. Nijssen then proposed a schema language based on a simplified version of the current DDL. In particular, repeating groups were to be eliminated from record declarations. Also, the concept of a (CODASYL) set was to be retained, but its definition

revised so that record types could serve as both owner and member in a given (CODASYL) set, and a set could have only one member record type. In addition, all sets were to be based on "matching" item values between items in member and owner records and were thus similar to sets which have been termed "inessential" by Codd.

Before reporting the debate that ensued, let us consider points 1) and 2) above in more detail. As evidence for 1), Mr. Nijssen cited a paper by Martin Huits of Philips Electrologica. This paper, "Requirements for languages in data base systems," was presented later in the week. Huits demonstrates some "real-life" problems which he feels are better solved procedurally than descriptively. The problems were coded in DSL- α and a procedural language not unlike ALGOL. As the problems become more complex, the claim is that the procedural solutions become simpler than the DSL- α solutions. Regarding point 2), Christopher Earnest presented a paper entitled "Selection and Higher Level Structures in Networks," which demonstrated a high-level query and update language based on an underlying network model of data. One interesting point brought out in Earnest's paper has to do with identifying "updatable" derived structures--those which have a unique reflection in the base structures from which the derived structures come. Earnest demonstrates how the conditions for updatability can come from the schema structure when CODASYL sets are used.

Returning to Nijssen's proposals, I think it is fair to say that much of the rest of the conference centered around the question of "how essential are essential sets."* On the one hand, Nijssen claimed the "matching item" approach was all that was necessary in the schema language and that set selection based, for example, on currency should be abandoned. Once one took this approach, Nijssen claimed it became possible to support any data model and thus one did not find oneself in an either/or (network/relational) situation. In addition, both the schema language and certain DML functions were greatly simplified, making them easy to teach and thus more accessible to users.

Nijssen's principal debating adversary was "Tax" Metaxides. He used the following example



* Phrase suggested by W. C. McGee.

and the term non-information-bearing to mean the same as inessential.

1. In both cases, a programmer would have to supply E# and D#
2. Placing D# as part of the employee entity was bad data structure design. The proper model was

```
EMP(E# . . .)
DEPT(D# . . .)
R (E#,D#)
```

and this was precisely what the "information-bearing" diagram said. (He did agree that it may be necessary to add phrases to the schema language to make definitions of primary keys easier to state).

3. The lack of a distinction between the existence of a relationship and the value on which it is based may be harmful. Users will use the D# item in EMP even though it does not belong there. Thus it cannot be removed (and thus there is more data dependence).
4. Set Selection will be simpler when "matching items" are used, but only at the cost of reduced selective power.
5. Complex selection conditions (e.g., place in a set if and only if there have been 5 consecutive evaluations with a grade less than B) are difficult to express with only matching items.
6. Without information-bearing sets (and the DML INSERT and REMOVE statements), it is
 - hard to build queues
 - hard to load the data base (consider the case where there are cycles in the schema)
 - hard to write system program utilities for the data base
 - hard to allow factoring of information which may be what some users want to do.
7. Placing the primary key of the owner within the member was but one of several implementations of the idea of a set and that other implementation methods were equally valid. So long as the system could recover the relationship

```
R(D#,E#)
```

it was irrelevant how the implementation had been done.

Thus Metaxides felt that perhaps extensions to the schema language may be necessary to more easily accommodate, say, a

relational view, but that it did not follow from this that other facilities should be removed.

Mr. Nijssen responded with several programming examples which he felt demonstrated the increased clarity and data independence that was possible if one took a "matching item" approach to the set selection. There was much discussion concerning the merits of his programs relative to other possible programs which would also remain invariant under changes to the schema.

Other papers presented during the week pointed out various flaws in the current schema DDL and proposed various extensions. Among these were discussions of data base procedures, proposals for revision of DYNAMIC sets, and proposals for a more level-structured data definition process, with various "physical attributes" only being declared and manipulable at the lower levels of data description. There were also presentations by D. Tsichitzis and M. Senko, each comparing various data models. T. Steel summarized important concepts of the forthcoming ANSI/SPARC report concerning the potential for standardization of various aspects of data management systems.

The conference closed with a series of resolutions that will be carried back to the DDLC, along with formal proposals for schema language revisions. Among these resolutions were the following:

- 1) Allow Owner and Member records of same type in a set declaration.
- 2) Eliminate repeating groups.
- 3) Allow global search keys in a record declaration as well as intra-set search keys.
- 4) Allow declarations of candidate and primary keys and revise Set Selection to allow for selection based on matching items.
- 5) Allow only one member record type in a set type.
- 6) Allow cardinality constraints in a set occurrence.
- 7) Allow a more powerful selection capability within Set Selection.
- 8) Allow for VIRTUAL sets and a revised definition of DYNAMIC sets.
- 9) Clarify and possibly extend the capabilities of SOURCE and RESULT clauses.
- 10) Work toward higher level operations in the DML.

At the close of the conference, participants agreed that the time had been well spent. It was one of the first extended discussions of why the CODASYL DDLC included or has retained

various features. DDLC members also commented that they found the feedback regarding various features valuable.

Proceedings of the conference will be published by North Holland Publishing Company, with B. C. M. Douqué and G. M. Nijssen, editors.

* * * * *

SIGMOD in Michigan

Under the sponsorship of the Metropolitan Detroit Chapter of the ACM, a new SIGMOD has sprung up. The new local SIG has been organized by Etelle Grinoch, who is also the founder of the New York area SIGMOD (originally known as DATABASE MAP). The Detroit SIGMOD has taken off with a modest membership of approximately fifty people and a program which includes: a study group on Data Dictionary/Directory, a tutorial presentation on the Codasyl DBTG report, and a panel discussion featuring the leading data base management systems. For information on membership and schedule of events, please write to:

Etelle Grinoch
DBA Corporate Manager
The Bendix Corporation
Bendix Center
Southfield, MI 48076
or call her at (313) 352-5357

* * * * *

FURTHER SUGGESTIONS FOR THE REVISED NEWSLETTER NAME

Database Management Review

Database Management Bulletin

* * * * *