INFORMATION SYSTEMS FOR EVALUATION RESEARCH:

A CASE STUDY

by

Jeffrey H. Loesch
Evaluation Unit
Crime Control Planning Board
State of Minnesota

July 1978

INFORMATION SYSTEMS FOR EVALUATION RESEARCH:

A CASE STUDY

Monitoring criminal justice projects and evaluating the effectiveness of programs in the criminal justice area often entails assembling large amounts of data. This is particularly true of client-oriented projects, in which individuals having widely varying characteristics receive specific and quantifiable services. Properly collected and organized, this data can provide excellent material for evaluating the effectiveness of both individual projects and collective strategies of treatment. There are, however, many difficulties associated with the management of such data, including:

1) Sheer size - The amount of information is nearly impossible to handle manually, and requires considerable resources even when computerized.

2) Reporting schedules - The timing of events is such that data is collected at irregular intervals, yet all information supplied to date must be available, and some information must be collected in timely fashion to assure analytical validity (e.g., follow-ups).

3) Need for frequent updating of information - As clients are accepted, terminated, convicted, or followed-up, new data must be collected and stored with other information already available for a given client.

4) Need for individual and statistical information both in standard reports and on an immediate *ad hoc* basis.

To overcome these difficulties and to improve the quality and availability of data collected, the Evaluation Unit of the Minnesota Crime Control

Planning Board (the state criminal justice planning agency), implemented a computerized information system, dubbed CODE, for client-oriented data for evaluation. This may be the first system designed explicitly for continuing evaluation research.

Since 1972, the Unit has collected data concerning the clients of all grant-funded projects that are client-oriented (youth service bureaus, counseling agencies, legal services, residential juvenile programs, halfway houses, diversion projects, community corrections, etc.). Data collected includes demographics, offense histories, referral sources and reasons, amounts of various services provided, offense record during project participation, agency referrals and their ratings, reasons for termination, subsequent record at periodic follow-up, and subsequent conviction. From the beginning of collection, this information was put in machine-readable form. Unfortunately, no satisfactory method was developed for accessing the data. When evaluations were done, a copy of the data for the relevant projects was extracted from the master file and analyzed using SPSS. There were several problems with this approach:

1) The data was not systematically checked for logical consistency as it was placed in the master file. Thus, any "cleaning" was done just before the analysis, when the opportunities for data recovery had passed. Furthermore, corrections that were made did not find their way back into the master file.

2) SPSS could not deal directly with the non-rectangular structure of the data as it existed in the master file. If an analysis was to associate arrest records with demographic and treatment history, a file had to be "dummied up" so that each client appeared to have as many arrests as the worst recidivist. The extra arrest records, comprising as much as 90% of the file, contained missing data codes that dropped out when statistics were computed. This procedure was cumbersome, inefficient, and expensive.

3) As the sequential master file grew, the expense of updating it and copying information from it grew in proportion.

4) In addition to providing no good means of data checking or statistical analysis, the masterfile system did not provide any feedback to the projects that were submitting the data. Requests for client-level or project-level data could not be satisfied economically, either in terms of computing or personnel costs.

As the size of the master files increased, information became decreasingly available.

In response to the difficulties encountered, the Evaluation Unit explored ways to improve data management. Many of the difficulties had been addressed by various commercial database systems.[1] These systems seemed to solve the update and individual-level availability problems, but they did not adequately address the need for statistical information, particularly for ad hoc statistical queries that are common to evaluation research. For most commercially available systems, the solutions to this problem resembled those available in the masterfile system: customized programs that lacked flexibility or provided highly inefficient interfaces to SPSS.

Nevertheless, a widely-used commercial data management system was initially selected, largely because of its availability on the computer

---

[1]For a general introduction to database management concepts, see: James Martin, Principles of Data-Base Management (1976), or Gordon C. Everest, Database Management: Objectives, System Functions, and Administration (Forthcoming, McGraw-Hill). For a brief history of the evolution of data management technology in business, see articles by Fry and Sibley in: "Special Issue: Database Management Systems," Computing Surveys, March 1976. Considerations in selecting a database management system are discussed in: CODASYL Systems Committee, Selection and Acquisition of Database Management Systems (1976).

used by the agency. This system, called System 2000,[2] provides extremely sophisticated update and individual-level data access capabilities. Unfortunately, it lacks statistical capabilities, beyond the most rudimentary (simple frequencies, counts, sums, and means) and the access capabilities involve a large initial investment in computing at the time data enters the database. To overcome the lack of statistical capabilities, a customized statistical reporting program was written that accessed the System 2000 database directly. The question of ad hoc statistical analysis was left unanswered, presumably to continue more or less as it had in the masterfile system. The difficulties of the custom-report strategy became apparent as soon as the program was finished: it did not meet most of the needs of planners, evaluators, researchers, or the projects themselves. Changes were difficult to make, and the users could not clearly anticipate their needs. Additionally, the costs of loading a 5,000-record pilot data set into the system were on the order of eight cents per record. Although the loading program was later "tuned" to improve efficiency, the projected costs still appeared far too high for data files totaling more than 100,000 records. This was particularly true in view of the fact that as data collection evolves, reorganizations are periodically necessary. In the case of System 2000 and most other systems, these reorganizations often require complete unloading and reloading of all data.

---

[2]System 2000 has been marketed and maintained by MRI Corporation since 1970. Versions are available for IBM 360/370, CDC 6000/Cyber, and Univac 1100-series machines. All references refer to Versions 2.6 and 2.7 of the CDC implementation.

In the light of lessons learned from the System 2000 pilot project, a new solution was sought. A system first released in December 1977, Scientific Information Retrieval[3] (SIR) was examined and seemed to offer many features not available elsewhere. Among the most important aspects of SIR are:

1) It is specifically research and statistics oriented.

2) It incorporates rigorous data-checking capabilities.

3) It has an SPSS-like language that greatly facilitates the access of researchers to the data.

4) It can deal efficiently with variable-size cases that are updated irregularly.

5) It can write directly SPSS and BMD save files for application of statistical routines not included in SIR.

6) It has online statistical and individual-level query capabilities.

7) It has an excellent report generator for writing standardized reports.

8) It is economical. Loading costs were 8% of System 2000. Retrieval costs were approximately equal. System 2000 was more efficient for retrieval of single individual-level items, but not for retrieval of all data on a given client or project.

While it is not the intent here to present an explanation of data structures in any detail, an important point must be made concerning databases with a major research component: The most demanding forms in which data is to be retrieved largely determine the structures by which data should be stored. Research demands on a database (as opposed to

---

[3] Scientific Information Retrieval is marketed and maintained by SIR, Inc. Primary documentation is contained in Barry N. Robinson, et al, SIR Users' Manual (1977). SIR is currently available for CDC 6000/Cyber machines, with a version for IBM 360/370 machines under development.

production and item query and update demand) are often the most difficult to fulfill and likely to be the least efficient and most expensive. This is because research needs are largely ad hoc, and hence very difficult to anticipate.

The object of utilizing logical data structures for research as well as for other uses is to minimize the total cost of retrieving any specified set of information. Hence, human as well as machine efficiencies must be considered. If ad hoc statistical requests can only be met by extensive programming, these costs must be included, whether or not the programming is done by research personnel. Furthermore, the lost opportunity costs occasioned by time delays in fulfilling information needs are also a consideration. These arguments apply equally to sequential masterfile systems or to the newest and most sophisticated structures for data access. This is not to say that the researcher should be concerned to any great extent with data access methods underlying his/her logical view of the data. However, the mapping of logical to physical access paths must be reasonably cost-efficient in the context of other desirable features of the system to be used.

To illustrate the above points, the data structures used in the CODE system may be considered. Data is available at the program, project, and client level. Programs are not in themselves formal entities, but may consist of the set of projects utilizing a particular treatment strategy or having a particular category of clientele. Projects are the grant-receiving entities. Evaluations may be either program- or project-level in their scope. Furthermore, they may be comparative, necessitating the retrieval of similar data from diverse project types. At the

client level, information is captured from the time the client enters a project until as long as three years after he/she has left it. This information is reported and entered into the database shortly after intake, again at termination, and at from one to four periodic follow-ups. Additionally, other information concerning convictions is kept for some categories of projects. This data is entered at any time after intake. Because of these chronological dependencies, data must be checked not only against other data in the same records, but for chronological consistency. Thus, for any existing client, data must be retrived to validate data that is to be entered. For continuous data collection, such dependencies are more often the rule than the exception. Failure to perform such validation may lead to striking anomalies which are discovered only during evaluation research analysis.

Figures 1 and 2 illustrate the logical structures of the two databases that comprise the CODE system.[4] The basic unit of analysis is the project-client, organized such that project-level aggregation is most easily performed. To permit the use of common report-writer functions and standard statistical retrievals, data unique to particular project types (e.g., Restitution in database 1, Supplementals in database 2) was stored in separate records. This practice also had the desirable effect of eliminating the storage of irrelevant variables for some projects. The record structure used by SIR occasioned representation

---

[4]The diagramming conventions used here are modified from Everest, Database Management. Connecting lines indicate association only, and not direction or dependency. For these figures, all records are dependent upon the above-connected record.

of some repeating groups as multiple variables rather than as multiple records, with the consequent support of some "not applicable" codes. The levels at which repeating sets of variables were broken down into separate records of a given type were chosen to conform to known characteristics of the data. Referrals, for example, tended to cluster--a client having one usually has two or three--but the modal number of referrals is zero. Hence, the referral record contains up to five referrals, but exists only for clients having at least one.

Figure 1: Structure of CODE Database Number 1

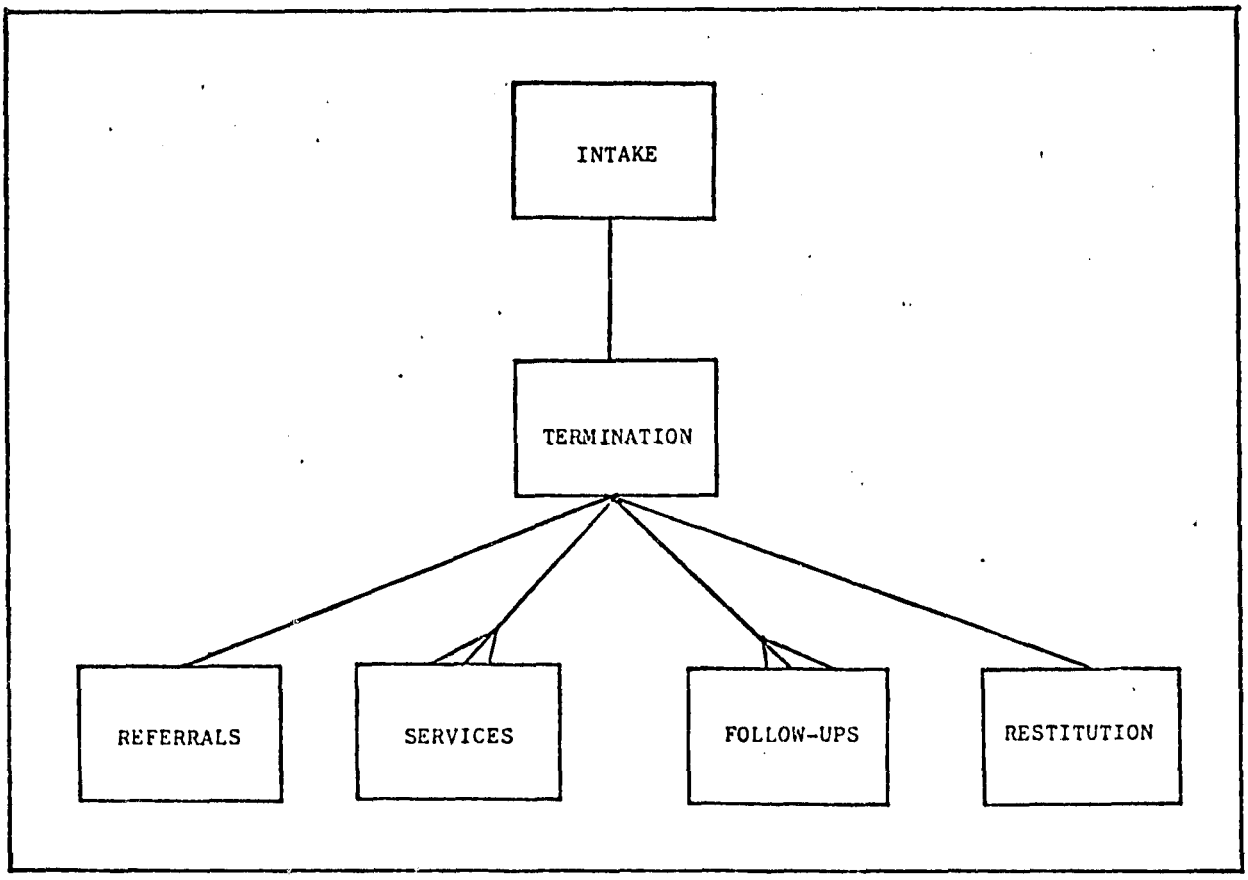
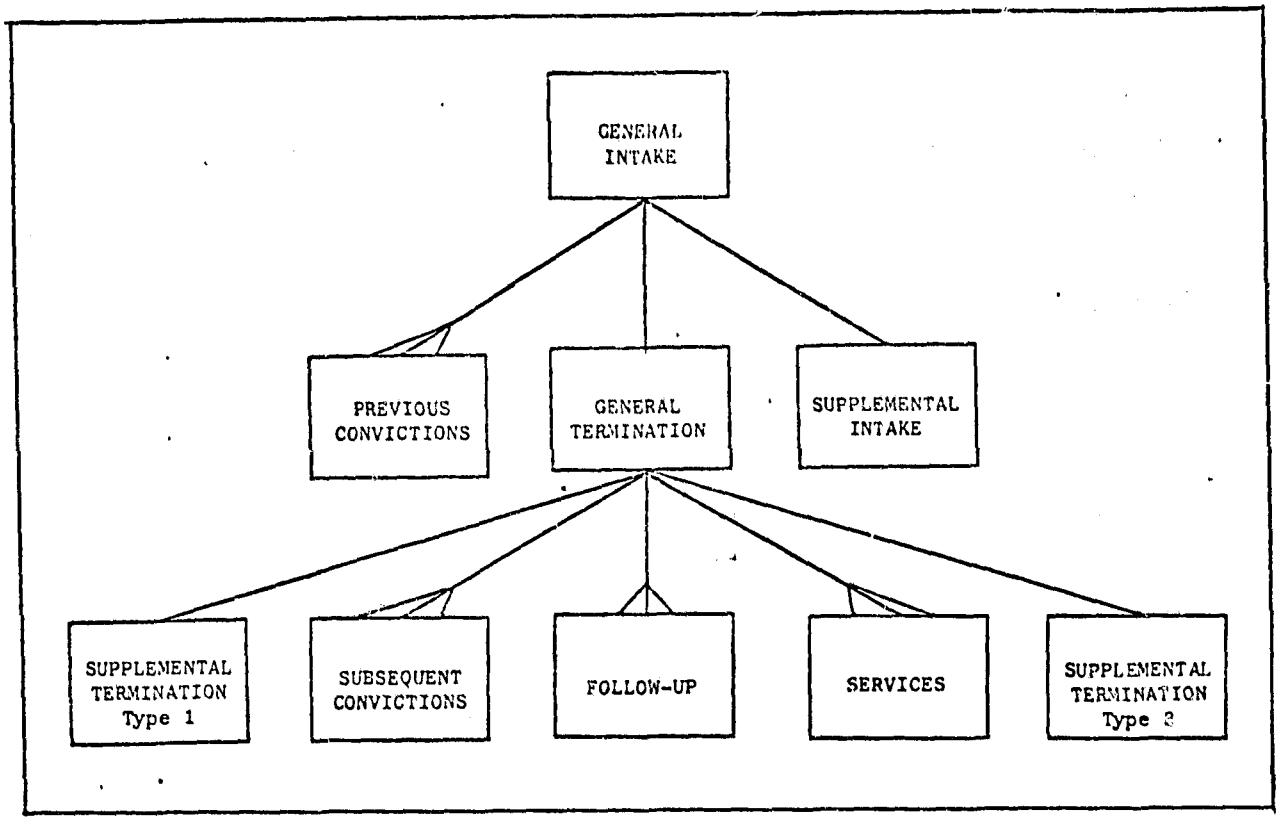


Figure 2: Structure of CODE Database Number 2



While System 2000, with its variable-by-variable ("component") structure completely eliminated the support of non-occurring fields, the complexity of the resulting structure occasioned other inconveniences of input and retrieval. Furthermore, the size of the System 2000 files was actually considerably larger than the corresponding SIR files, despite the fact that the SIR files contained a significant proportion of non-occurring fields. This is not to say that the overhead of building and storing the System 2000 structures (inverted lists) is never justified for statistical research--only that it did not appear cost-effective in this situation, because the access methods could not conveniently be harnessed to the required statistical procedures. The access methods provided by SIR, though less sophisticated than those

offered by System 2000, were appropriate for the required research applications and provided equally efficient access at lower loading costs.

The CODE project as implemented has already begun to bear fruit. As soon as one database was loaded, reports on active clients and overdue and soon-to-be-due follow-ups were mailed to all projects, with promises of more information as the system progressed. The quality and completeness of data submitted by projects began to improve immediately. A fairly detailed statistical report was mailed to all projects. With this type of feedback promised on a quarterly basis, incentives for complete and accurate reporting increased. Follow-up calendars, ordered by due date, help eliminate missed or late follow-ups that compromise the evaluation of project and treatment effectiveness. Summary reporting statistics are compiled monthly or on demand, so that systematically delinquent projects may be identified. Copies of the quarterly project statistical reports are provided to evaluators and state and regional planners, as well as to the projects themselves. Tnese provide the basic information for beginning an evaluation, and serve as a springboard for further ad hoc statistical analysis of projects or groups of clients receiving certain treatments or having certain characteristics.

The CODE information system as it is currently designed has thus far been successful. It overcomes all of the difficulties encountered in the masterfile and System 2000 designs. It is still too soon for a detailed cost-effectiveness evaluation of the system. A few casual observations can be made, however. Routine operating costs for the

system appear to be less than those of the old masterfile system, especially when the costs of the previous methods of analysis are included as a previous operating cost. The use of these methods has been eliminated by the new information system. Likewise, it is likely that development costs, at least 80 percent of which are attributable to personnel costs, are offset by long-run savings in analytic costs, which are also primarily personnel costs. The only major additional cost is the running of full quarterly statistical reports for all projects. This cost is hopefully offset by the value of the additional information provided. It is difficult to assign benefit values to improved evaluations and better relations with project staffs. Valuation of data quality might also be difficult, but improvements in error rate and completeness can definitely be measured.

While the system thus far appears successful, it is unlikely that it will violate the "Iron Law of Information." ("The demand for information always expands to exceed the capacity to provide it.") Nevertheless, it is likely that the system will continue to evolve to provide more accessable information for evaluation of a wide variety of client-oriented programs and projects.

END