

4829Y

COURTRAN: A MODULAR MANAGEMENT INFORMATION AND RESEARCH SYSTEM FOR COURTS

Joseph L. Ebersole
Director of Innovation and Systems Development
and
James A. Hall, Jr.
Senior Systems Analyst
Federal Judicial Center
Washington, D.C.

INTRODUCTION

COURTRAN is the name given to a computer software system designed by the Federal Judicial Center to provide information support services for court management and for study and evaluation of court procedures and processes. Major features of the system are a language designed to handle court transactions, modular programming, an expandable file structure, and modifiable external dictionaries and tables for information elements. There are four versions of COURTRAN: criminal operational, criminal research, civil operational, and civil research. Each of these is a variation of the basic COURTRAN system using the techniques described herein.

The characteristics of the system were dictated by the special requirements for its operation and the research and development mission of the Center. It was designed to be operated in federal courts, by and under the control of local court personnel, to serve local case management information needs, and to provide local data required for input to the national statistics system. The criminal operational version has been implemented in two courts, the civil operational version is being implemented in one court this fall, and the civil research version will be used for a Center research project after implementation of the operational version.

Although our purpose here is to describe how the COURTRAN system operates and the functions it performs, we believe this description will be more meaningful if the design philosophy and the considerations which shaped the objectives of the system are first explicated.

Many existing computer systems required the user to conform to the demands of technology and thus make humans the servants. We were concerned with an approach which would make technology the servant of persons in the courts, and would allow parajudicial personnel to decide both the information products of the system and the way they would interact with the system. This posture affected both the nature of the software and

the process used to define the specific objectives and functions of the system. For example, we oriented the design to reflect case processes and stages in the life of a case instead of orienting it to clerical or statistics gathering processes. Thus, the system is not tied to existing record keeping procedures, but to the events which occur as criminal and civil cases progress through the courts. As a consequence, system operation results in a progressive reexamination of procedures, new insights into the information functions of court supporting operations and changes in procedures as a result of these insights.

Court administration is an inchoate discipline which may some day gain the status now given to public administration or business administration. Since a computer system for court applications should be a tool to help improve court administration one should not design such a system to be rigid and inflexible as if the final answers about the best techniques were known. There is much yet to be learned and a system should be considered experimental and should be used as a learning device as well as an operational device. It should be designed for change since change is the only certainty both in our society and in our courts.

With this general philosophy as a starting point, a participative design process — involving a design group consisting of administrative personnel from district courts and Federal Judicial Center staff members — was used to determine the types of functions which should be performed by the system and the types of products which should be provided. This is not the most efficient way to design a software system. It has all the disadvantages of the democratic process, but it also has some of the advantages. It allows the user to decide what he wants, insures practicality, and precludes many of the problems that occur when a system is designed in a technological ivory tower or when systems analysts try to impose their concept of needs on the user.

Use of a participative rather than a prescriptive approach does not change the fact that the design process for a court system will be different from more typical computer applications, where needs are clearly perceived, can be easily articulated, and can be stated with the precision necessary for automatic data processing. Normally, a special study is conducted in order to determine "requirements."

Defining requirements is relatively easy if one is developing a system to handle accounting or book-keeping activities or other typical business processes for which a precise set of rules and practices already exist. The situation is quite different when one attempts to define requirements for a less routinized, more dynamic set of processes. Also, the fact that the judicial system is to some extent a nonsystem creates unique problems. Supposedly, people in an organization all work to further one general objective and although there are opposing forces in existence, these are normally not intended to operate at cross-purposes. Business organizations, for example, strive for efficiency and attempt to systematize as many activities as possible to further this goal. But the administration of justice is, in some respects, inherently inefficient. The due process model is a purposive obstacle course and its inefficiency and, what some might consider its irrationality, provide major protection to individuals. The system designer has to rethink his concept of systems when he confronts courts' requirements. However, the objectives of both justice and efficiency can be served by designing software which provides better information on court transactions and case events so as to help judges and court administrative persons manage cases more effectively while identifying situations where injustices may be occurring.

FACTORS AFFECTING SYSTEM OBJECTIVES

The objectives of the COURTRAN system were shaped by the environment. The design group had to confront the hard realities of the deferred maintenance syndrome and its heritage of limited court budgets for developing and testing innovations. An objective which emerged from this consideration was that the system should not require a separate large staff of technically trained ADP people in each court. It should be operable, and its products usable, by existing court personnel. In conformance with this objective, the system was designed to simplify to the extent possible both input preparation and processing operation. Ease of input and operation is described as "transparency" to the user. This transparency, this relative simplicity, requires substantial complexity in the software design. Software complexity and user complexity are inversely related. If one designs a system which is easy to program, it will be difficult to use. If the systems specialist ducks the hard problems this merely transfers them to the user for solution. Technology then becomes the master, not the servant, of the user.

Even the largest federal courts do not now have the volume of cases necessary to support an on-site computer. Therefore, the use of rental time for both development and pilot operation was mandated. This meant the system should be designed so it could be operable on computers located reasonably close to district courts. In effect, the objective was to design a software package which could be taken to various courts and "plugged-in" to a local computer.

Although there are basic similarities, each court's needs are to some extent unique, and a realistic approach to design and implementation requires the system designer to recognize the diversity which exists among district courts. Each reflects the culture and traditions of the locality and especially of the local Bar, and this affects the specific practices and procedures of a court and creates inarticulated norms. There are also differences in the case "mix" reflecting the commercial and population characteristics of the district, differences in physical facilities, and the geographic concentration of judges. Furthermore, there are a number of different organizational structures in court clerks' offices. Each of these features causes differences in the types of information products which are most useful for administrative functions.

This diversity indicates that a rigid, completely uniform system would not serve the needs of various district courts. This is not to say that some degree of standardization is not possible. In fact, many key elements of administration are already standardized. But local operating procedures vary and a system must be able to mesh with them. Thus another system objective was that flexibility and adaptability be incorporated into the design. Like simplicity for the user, flexibility and adaptability require greater complexity in the software, more development time, and more problems for the system designer.

Another factor with which we were concerned was the cost of court studies which involve measurement and evaluation of court procedures which relate to the characteristics of cases and case events. Courts typically have limited capabilities for determining the cause of problems which may exist or for evaluating the effect of changes in procedures. Most court studies have suffered from the high costs of obtaining information from court records, the paucity of information therein and the unreliability of such data as is obtained. In fact, research project directors have often found their funds nearly exhausted by the time the data collection effort is completed. Analysis, which is the

most important part of any project, then suffers from lack of funds and lack of time. Because research is one of the missions of the Center, we established the objective that the system be a tool for both management and research.

A research project normally involves collecting information on case characteristics and event history in totality at some point in time after a case is closed. In contrast, in a system for management purposes, information related to cases and case events is captured as it is generated. A more important difference between management and research information is the greater depth usually required for a research project. Therefore, the design includes a structure which enables the handling of several hierarchical levels of information density, *i.e.*, both general data and one or more degrees of successively more detailed data. For example, one level of information density might consist of the fact of occurrence of key events in the life of a case. A more detailed level might involve more information about the characteristics of each event. Another level could involve breaking down each event into a series of subevents.¹ The capability to process different degrees of information density makes it possible to use the software for a special data collection effort or to add data about event characteristics to already existing computer files thus reducing the costs of special studies and

allowing them to be conducted by use of a processing technique with which court personnel are already familiar. In more typical computer software systems, the collection of additional data elements requires the system to be redefined and reprogrammed, a task which sometimes doubles the cost of the programming work when a change is required in a project.

Although there are differences among courts, there must be some degree of standardization if national statistics are to be meaningful. A core of standardization already exists in the form of the JS (Judicial Statistics) reports which are prepared monthly by each federal court and forwarded to the Administrative Office of the U.S. Courts for compilation. The totality of information included by a local court in its version of COURTRAN can be viewed as a mathematical "set." A portion of this information is required for national purposes. As illustrated in Figure 1, the data for JS reports can be considered a "subset" of the local court's data "set." Therefore, all data elements in this "subset" were standardized to conform to the requirements of JS reports. It was also decided to design the system so it would be able to automatically produce — and punch — this input. In this sense, and for this "subset" of data, the local court system is a satellite of the national system

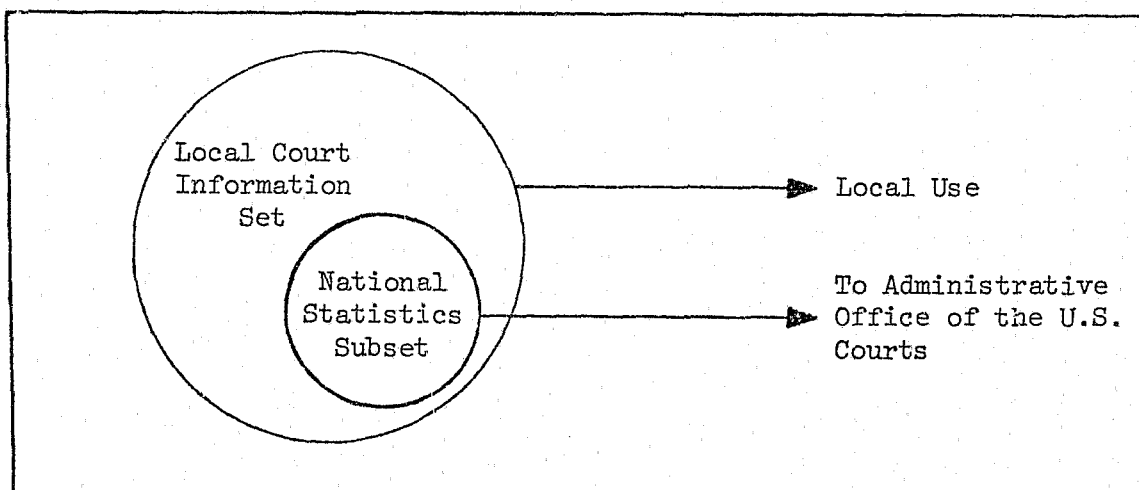


Figure 1. Local Information Set and National Subset

Because on-site equipment could not be justified, and because it was not possible to revolutionize paperwork and clerical procedures and replace the majority of manual tasks, initial priority was placed on the information and evaluation products of the system. The design group did, however, establish the objective of replacing manual tasks wherever feasible or possible. Since the inception of system operation, a number of tasks previously performed manually have been replaced resulting in cost savings which exceed the cost of computer time. A given manual task was not eliminated until it was proven that the COURTRAN prepared reports were more accurate. This criterion was met in every instance where a manual task was a candidate for elimination.

One danger sometimes not recognized by either users or system designers is that the implementation of a computer system may freeze office procedures and preclude the possibility of further change and improvement. This has been the fate of many organizations which have implemented computer systems. Even though the development and implementation of a computer system usually involves significant improvement in office procedures, this change should never be designed to be a final "fix" or solution. Therefore, another objective was to design the system to be paperwork-independent so as to preclude rigidity in procedures which would limit or inhibit other administrative objectives.

The introduction of a computer system in most organizations results in the emergence of previously unperceived needs and an enhanced awareness of problems in present procedures. The anticipation of this phenomenon dictates an evolutionary approach to computer system development and implementation, and a design which enhances rather than constricts this evolutionary process. Thus, another objective was to design the system to facilitate change and evolutionary development.

There is a major trend today toward on-line systems. Although more costly, they provide more responsive and more timely services. Traditional systems are batch oriented, *i.e.*, both input and output processes are performed on a periodic basis (usually monthly, weekly or daily) by grouping all data for the period and processing it in one batch. During the interim period, many events occur and many data needs arise which must await the time of the periodic batch cycle. The period can be reduced to daily processing cycles, but even then unforeseen questions may arise. The ideal situation would involve the capability to obtain immediate

response to information needs. An on-line system may provide the answer here. Therefore, another COURTRAN objective was to incorporate software design concepts which would make it possible to convert to on-line operation when a reasonable cost level is reached.

Another objective was to provide more refined court statistics. The courts have a public responsibility to achieve resolution of cases in the shortest amount of time consistent with a high quality of justice. Sometimes court statistics give a misleading impression of the effectiveness and efficiency of judges. In this sense, case statistics are often unfair to devoted, hardworking, overburdened judges. These statistics contain implicit inferences about the extent of, and the causes of, case delay. Therefore, two steps were taken to refine the present gross statistical measures. First, the system was designed to be able to account for all time periods during which the case is delayed by causes not under the control of the court. Second, the system was designed to provide statistics on a per-judge, not a per-judgeship, basis.

The first objective is applicable to both civil and criminal cases. There are numerous situations in which a case is in an untriable or "limbo" status. Criminal cases are untriable, for example, when a defendant is a fugitive; when he is committed for evaluation to assist in determining competency; when a case is delayed pending disposition of another case (in which the same person may be a defendant or the disposition of which will be determinative of certain matters); or when a case is delayed pending an interlocutory appeal or an appellate decision in another case. There are other situations in both criminal and civil cases where delay may not be controllable by the court or where delay may produce a more just result. This does not refer to attorney preparation time or necessary periods for discovery (which vary according to the type of case), but to situations where court action may be deferred for a period of time when it appears this will lead to a more fair result. Where such periods can be defined by an objective standard they should be so identified and recorded by a computer system. This suggests that a more accurate measure of case processing time is the elapsed time between filing and disposition *minus* the elapsed time during which the case is in an untriable status or during which a deliberate extension of time has been provided according to a rule or standard. Thus, a generalized technique for accounting for such time periods was incorporated in the system software.

The second objective, that of maintaining case statistics on a per-judge basis, is an automatic fall-out of a localized system since the system works with actual case assignments and does not compute more active judges than are in being. Thus, the measure of output is based on the actual number of judges, not the number of judgeships.²

During the design period, meetings were held with members of the law and engineering faculty at the University of Notre Dame who were developing a simulation model for criminal court processes under a grant from LEAA. Their innovative study, which involved an integrated project team of lawyers and engineers, used techniques which meshed with characteristics of the COURTRAN design. Their experience confirmed the usefulness of the event-oriented approach to software design and led to the establishment of another objective, viz., the ability to produce direct input for the Notre Dame LEADICS simulation model. With this capability, it will be possible to simulate changes in court processes without the extensive and expensive data collection and conversion efforts which have often plagued model simulation projects.

Our final objective was derived from developments in the Congress and the Judicial Conference Committee on Criminal Rules relative to speedy trials. Although there have been several Bills introduced in the Congress and several tentative versions of a proposed new Federal Criminal Rule, each contained some similar features. These were: preparation of a plan for prompt disposition of criminal cases; time limits during which specified events should take place; designated circumstances which could be the basis for extending these time limits; and, a means for reporting the status of cases. Several of the system objectives noted above fit in with the principles of the various speedy trial Bills and proposed rules; but in attempting to provide status reporting and monitoring information, another feature was added to the design. This consisted of a technique for establishing "suspense" periods for each stage of a case and a technique for computer generation of notices on cases which were exceeding or about to exceed these limits. As part of the adaptability features of COURTRAN, the software design provides for a method by which each court can set its own time limits based on its plan. In April 1972, Criminal Rule 50(b) was approved by the Supreme Court. Hopefully, the features designed into the system will be such as to support implementation of this new "speedy trial" rule.

COURTRAN CHARACTERISTICS

In order to achieve flexibility and adaptability COURTRAN was designed as a modular software system. Instead of writing a series of monolithic programs which are integrated into a fixed system, the modular concept involves development of many small programs (functional "modules"), each of which has a specific function. Control programs are then prepared which put together different combinations of modules to perform each major processing function. The advantage of modular programming is that one can put together different combinations of program modules and create different systems. Also, when a change is required, it is usually only necessary to reprogram one or two modules instead of reprogramming a total subsystem. This is one of the primary techniques used to achieve flexibility and adaptability. In addition, it results in greatly reduced costs when changes are required.

Another major design characteristic is the use of external dictionaries and tables. Judges' and lawyers' names are of course different in each court, names of bondsmen and correctional institutions are different, case events are often different or given different names, and the type of reports are sometimes different. In addition, civil case events and identifying features are different from criminal cases. The use of external dictionaries and tables allows a programmer to make changes easily to the system, to adapt it to each court, and to change the types of case data which it will handle. The implication of the adjective "external" here is that a programmer can effect these changes through revisions to the dictionaries and tables without making changes in the internal programming logic.

Possibly the most unique characteristic of the COURTRAN software is the use of a transition matrix structure for defining the events – and their effects – which occur in the life of a case. Transition matrices for Markoff Chains are well known to Operations Researchers. The COURTRAN transition matrix does not have all the characteristics of a mathematical transition matrix. For our purposes it is *inter alia*, a technique for symbolically representing case events and stages and can be viewed as a simplified flow charting technique. Figure 2 illustrates a sample portion of the flow of criminal cases in a court as this might be represented in a flow chart and then illustrates the same partial flow using the transition matrix structure. This illustration is not meant to be complete, but is used merely to describe the technique.

In the transition matrix, each cell represents a transition from one stage to another or a holding action. For example, the return of an indictment or information are two events which move a case from outside the court into the pretrial stage. The pretrial stage in Figure 2 is thus defined as the period between the filing of the case and the beginning of trial, the entry to a guilty plea, or the dismissal of the case. As another example the plea of guilty represents a transition from the pretrial to the presentencing stage which on the matrix is shown as a 1,3 transition.³ Guilty pleas can also be entered during trial, and this event is shown by the mnemonic "PG" which appears in the 2,3 cell in Figure 2. Events which are shown in the cells along the diagonal do not move the case to another stage and events in cells below the diagonal move the case "backward," i.e., a return to a previous stage. (See Appendix for examples.) Some examples of case dispositions are shown in column 4. For example, the "D" in cell 1,4 indicates a dismissal which occurred during the pretrial stage. The "D" in cell 2,4 indicates a case which is dismissed after the trial began.

In COURTRAN, a matrix can be broken into as many as 14 stages. Each court can define its own stages and the mnemonics which it wishes to use. The size and structure of the matrix depends upon the degree of detail desired by a given court and the nature of the information products which it wants from the system. Our experience has shown that this technique is easier than flow charting. Also, although it is sometimes difficult to develop a coded representation of a flow chart in a computer program, it is relatively easy to develop internal program codes using the matrix structure. For example, the "PG" and "G" mnemonics shown in transition cell 2,3 are coded internally in a computer as 2,3,1 and 2,3,2. (The last digit merely designates the first mnemonic in a cell, the second, etc.) The external dictionary then tells the system software that code 2,3,1 means a guilty plea entered after the trial has begun, and that code 2,3,2 means a defendant was convicted as a result of a trial.

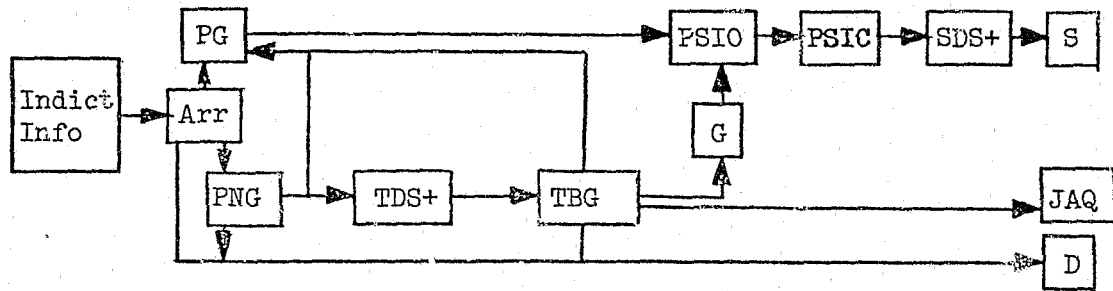
Prior to COURTRAN implementation the matrix technique is used by a court to describe and document its major procedural steps. As a result of going through this exercise, the pilot courts have found that their existing clerk's office procedures sometimes do not properly provide for all court events. Use of the matrix also shows where existing forms should be changed in order to provide more effective paperwork procedures.

Use of mnemonics is another key characteristic of the COURTRAN system. Their use greatly reduces the cost of key punching and the cost of computer processing, they are easy to learn since they are based on the word or phrase which describes the event, and they constitute the lexicon of the COURTRAN language.

Figure 3 shows a simplified system flow. The major steps of preprocessing, file updating and report generation are common to most computer systems. However, in the COURTRAN system each of the steps consists of combinations of program modules which vary from court to court and from civil to criminal applications. Every step in the flow uses one or more of the dictionaries and tables. For example, at the preprocessor stage, every event is checked to assure its validity and the type of case, name of judge and location are checked to be sure they are valid. The master file edit and update programs use the mnemonic dictionary for logical checks to assure conformance with the rules established when the matrix was designed by the court and to obtain various items of standardized data. The report generator uses the dictionaries and tables to obtain the narrative for each code, thus converting from computer-readable to human-readable symbols for printouts. Names and addresses of attorneys who practice in a given court are maintained in a separate subsystem. This requires more than a table since changes are frequently made as new attorneys appear, since that single subsystem interfaces directly with both the criminal and civil versions of COURTRAN, and since it is then available for use in systems designed to handle the engaged counsel problem. The names of assistant U.S. attorneys are maintained in a prosecutor table because addresses are not necessary, changes are less frequent, and the total number is much lower than the total number of private attorneys who may practice in a given district.

In addition to keeping track of events, COURTRAN also has provisions for scheduling information and for monitoring time periods which are established for case stages. Two scheduling mnemonics (for trial and sentencing dates) were shown in the matrix in Figure 1. As mentioned above, each court can set time limits for various stages. These limits create suspense dates and if another event has not occurred within the established time period, the case will appear on an exception report. Where a scheduling mnemonic is used, the date set for the event is punched immediately following the mnemonic. When a continuance occurs, the date to

Figure 2 Comparison between flowchart and transition matrix structure for a sample portion of a criminal case flow in a court.



	Out 0	Pretrial 1	Trial 2	Pre- Sentencing 3	Disposition 4
0		Indict Info			
1		Arr PNG TDS+	TBG	PG	D
2				PG G	D JAQ
3				PSIO PSIC SDS+	S
4					

INDICT - Commenced by Indictment.
 INFO - Commenced by Information
 ARR - Arraigned
 PNG - Not Guilty Plea Entered
 TDS+ - Trial Date Set (Plus Date)
 TBG - Trial Begun
 PSIO - Pre-Sentence Investigation
 Ordered

PG - Guilty Plea Entered
 G - Tried and Convicted
 SDS+ - Sentencing Date Set (Plus Date)
 D - Dismissed
 S - Sentenced
 JAQ - Judgment of Acquittal
 PSIC - Pre-Sentence Investigation
 Completed

which the event was continued follows that mnemonic. If a hearing or trial is continued but no date is given in the order, the case will appear on an exception report after 30 days; otherwise, the case appears on an exception report only if the scheduled date has been passed and the event did not occur. In some instances, this will not be because a trial or hearing was postponed, but because there was a deficiency in clerical operations. Thus, the first check made by the clerk's office after receiving an exception report is to make sure there was no error in data capture. Therefore, this is also a method for assuring accuracy and completeness of court records.

The capability for handling several hierarchical levels of information is implemented via the use of mnemonics. For example, we originally used only one mnemonic ("D") for dismissals in criminal cases. In order to be able to distinguish dismissals because of superseding indictments from those occurring as a result of a motion by the prosecution or defense, or *sua sponte* by the court, four additional mnemonics were created: DC — dismissed by the court; DD — dismissed on motion of the defendant; DG — dismissed on motion of the government (but not because of a superseding indictment); and DS(+) — dismissed due to a superseding indictment. The plus sign following the last mnemonic indicates that the case number of the superseding case is entered with the mnemonic. This change to a more detailed level of information allows more refined statistics on dismissal rates which will show the composition of a statistic that is often available only as a gross percentage. The creation of four new elements of information is accomplished by adding the four mnemonics to the dictionary without any reprogramming or changes in the internal file structure.

Similar changes can easily be made for other data. For example, there are mnemonics which record the occurrence of a continuance. By adding the letters D and G, and thus creating two new mnemonics, data can be available which shows whether defendant's counsel or government counsel have asked for the continuance. Then, if the number of continuances is excessive, the COURTRAN output will indicate where efforts should be focussed to reduce the seriousness of the problem.

The COURTRAN preprocessor was designed for "free-form" keypunching. Initial case identification information must be entered in a fixed sequence, but the card location and the length of an item are variable. The keypuncher merely

punches an item, skips two spaces, punches the next item, skips two spaces, etc. Keypunching for events occurring after a case is filed is even simpler since the system produces pre-punched cards containing the case number and defendant name and/or number. The keypuncher merely inserts the card in the machine, skips the pre-punched case identification information and punches in the mnemonic. Where several events occur for the same case, the keypuncher merely skips two spaces between each mnemonic representing these events. Experience to date has shown that training of a keypuncher requires at the most one day and this is for people who have never before seen a keypunch machine, but have had some experience with typing.

Accuracy of data has always been a problem in most court information systems and for most research projects. Therefore, the COURTRAN software contains a multitude of validation and self-checking functions to insure data accuracy. When information input does not meet a series of tests which use both the external tables and dictionaries and numerous logical routines, special messages are printed which explain the nature of the error. The system also contains complete audit trail capability so that any type of error can be traced to its origin.

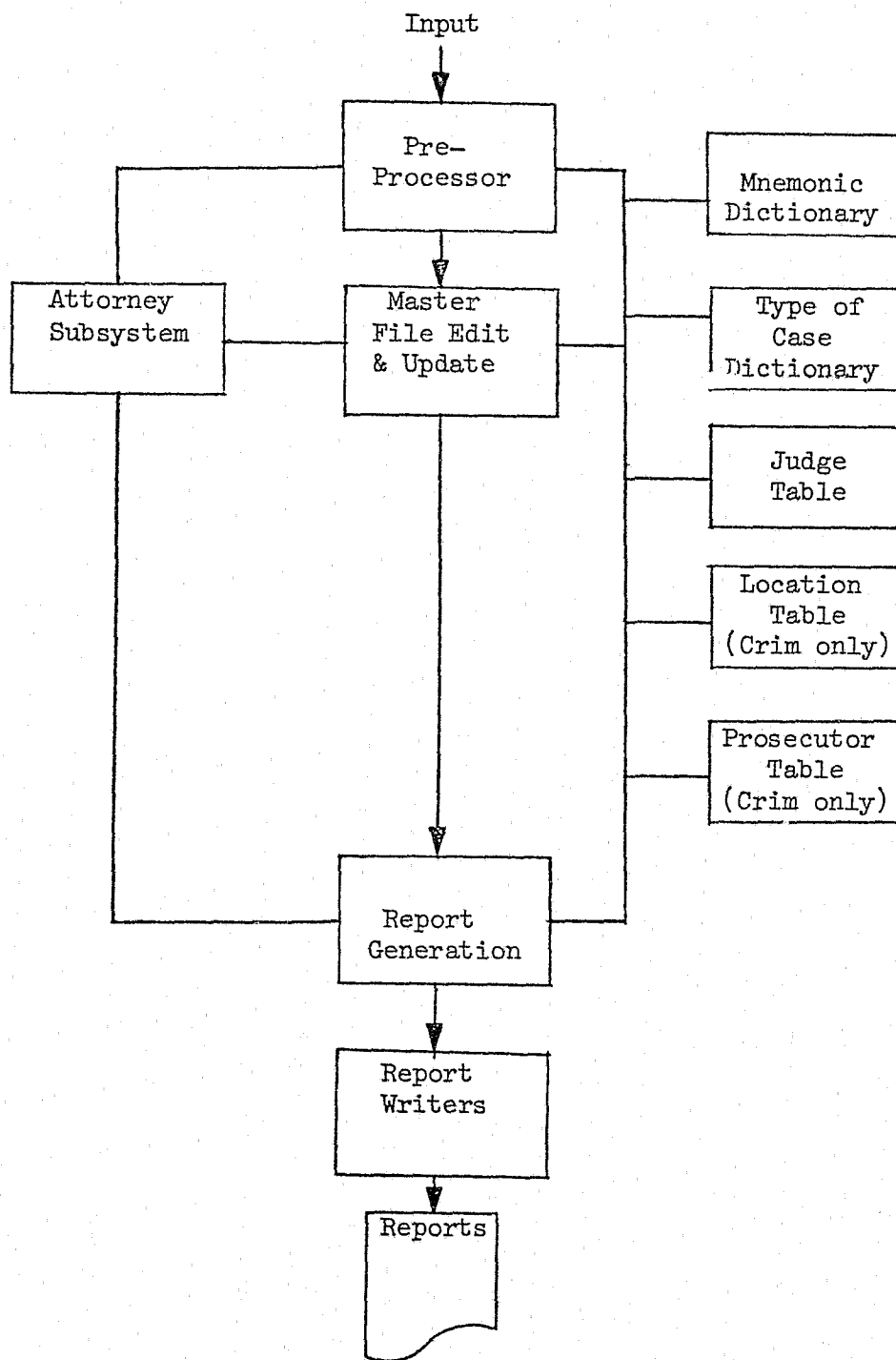
LINGUISTIC AND TECHNICAL CHARACTERISTICS

COURTRAN is, in effect, a high-level programming language which interprets and executes statements derived from source documents in the court. The linguistic structure consists of a *source* language, to be used by humans and a *target* language to be used by the computer. The COURTRAN translator converts statements in the source language into procedures defined in the target language. These procedures are then executed by the COURTRAN system.

COURTRAN source language consists of two kinds of statements: Control statements which identify for the system the input which is to follow, and source statements which provide the actual input. Control statements always begin with a dollar sign (\$) followed by a keyword, and may contain one or more additional parameters.

Source statements are composed of three grammatical elements: *subjects*, which identify a unit of judicial effort; *predicates*, which state what happened to the unit; and *modifiers*, which provide differing degrees of information about a *subject* or *predicate*.

Figure 3. General Systems Flow, Civil and Criminal



The *subjects* in the COURTRAN criminal system are a case, which is identified by a docket number, and a defendant, identified by at least a name and a sequential number related to the docket number. The sequential number is "1" for the first or only defendant in the case, and "2" through "N" for subsequent defendants charged in a single indictment or information. The *predicates* consist of a series of mnemonics representing case events, or changes in bail status or defendant location. The *modifiers* provide related or additional information about a case, a defendant, or an event. Source statements must contain a subject and at least one predicate. They may contain multiple predicates and modifiers.

The COURTRAN translator parses control statements and source statements, calls into service the appropriate parts of the COURTRAN system, translates the subject into an access key and retrieves the case record, interprets predicates and modifiers, and posts events and related information to the record.

Syntax for control and source statements is defined by a table in the translator containing the minimum number of items required in the input statement, the maximum number of items permitted, a list of verification subroutines, one for each item, and a list of error messages, one for each item. Syntax for the target language is defined by the matrix of mnemonics, designed by the user, which is implemented in the system as the mnemonic dictionary of predicates and modifiers, and by modules which check the validity and consistency of interpreted statements and define house-keeping functions implicit in certain mnemonics.

Each "word" in the mnemonic dictionary is defined by a simple card input. Columns 1 through 22 of this card contain text which will appear on reports. This text may contain edit symbols which indicate to the report generator that some modifier may apply, and these symbols cause text associated with the modifier to be inserted. Columns 23 and 24 must be blank. Starting in column 25, the definition of the mnemonic appears in free form, each field separated by at least 2 blanks. The fields necessary to the definition are: (1) the mnemonic, which must be from 1 to 4 alphabetic characters, (2) a two-digit numeric field containing the matrix coordinates of the mnemonic, (3) a 1 or 2 digit hexadecimal field containing the path number associated with the mnemonic, and (4) a field defining the suspense parameter associated with the mnemonic. The suspense parameter may be an alpha-

betic "V", to indicate a variable suspense parameter obtained from a modifier (e.g., a scheduled date), or a decimal value to indicate the maximum number of days after which a defendant will automatically appear on an exception report if no further action has occurred. A fifth definition field is optional, and identifies the kind of modifier which must or may be associated with the mnemonic (e.g., a date, a related case number, or a time period). After the matrix and mnemonic dictionary have been defined, an entry may be deleted, or may be changed by entering a new definition with the same mnemonic field, or new entries may be added.

The mnemonic dictionary is used by the input preprocessor, master file editor, and the master file update program to verify that a predicate is defined, and that any required or optional modifiers are present. It is also used by the master file editor and the update program to post events, update the suspense field, and store related information. The update program also consults this dictionary in conjunction with the designated record to verify the consistency of a transaction to be posted. The report generator consults the dictionary to retrieve cases by current status, or by the occurrence of a specified event anywhere in the history of the case. The case history lister, which provides a complete event-by-event listing for all cases, uses the dictionary to retrieve mnemonics from their internal form.

MODULAR STRUCTURE:

The number of modules or basic "building blocks" is approximately 500. These can be classified, for purposes of analogy, as executives, supervisors, and clerks. Each is a separate control section with sometimes specific and sometimes very general functions. These can be combined in many different ways. One subset of these combinations is the COURTRAN criminal system, another is the COURTRAN civil system.

An example of an executive component is the control module for the input preprocessor and master file update. By inserting 2 or 3 new tables to be called into this structure, we can redefine syntax for control statements and source statements. This executive calls upon various supervisors to check grammar and syntax of control statements and source statements. After checking is completed, the executive determines whether the statement is acceptable or needs modification.

One of the supervisors called upon by the control executives for the preprocessor and the master file update is responsible for verification of all items in the statement. This supervisor has access to a series of clerk modules, each of which is responsible for detecting errors or irregularities in a specific item. The supervisor reports back to the executive the findings of the individual clerks.

An example of a simple specific clerical task is the examination of the docket number for validity. This specific clerk must be replaced if a different form of docket number is to be used. The D.C. District Court, for example, uses a number in the form 1234-72, while the Illinois Northern District Court uses 72CR1234. Obviously, the D.C. clerical routine would reject a Chicago docket number, but equally obvious, the substitution of a Chicago clerk-module at this point in the process remedies the difficulty.

REPORT GENERATION TECHNIQUES:

Report generation programs, which form an interface between the system and the independent report writer, greatly simplify the programming for additional reports. The report generator reads in, or determines from tables, criteria for selection of records from the files, extracts the required information, and explodes internal codes to text, which is then written on a file as input to an independent report writer.

The report writer itself consists of an executive program which can read in report headers from cards for one-time reports or from disk files for recurring reports, determine which report is to be written, load the appropriate formatting and printing program, and pass text records to it. The actual programming for a new report consists only of instructions to print column headers and detail lines. Subroutines are provided for printing of page headers and for page numbering.

STRUCTURE OF MASTER FILE:

The basic unit of the master file is an 88-byte record. In the criminal system bytes 1 through 5 contain a key which is used to retrieve records, bytes 6 through 8 contain control flags, and bytes 9 through 88 contain data, usually in compressed and encoded form. The record key is composed of a docket number, a sequential number called the record or defendant number which identifies defendants within the docket number, and a sequence number, which is used for control pur-

poses, as will appear below. (See appendix for master file layout.)

Record number zero identifies the case control record which contains data pertaining to the case as a whole. Thus far, this data is confined to a single 88-byte record with sequence number zero. As the need arises, other 88-byte increments for case-related information may be inserted with sequence numbers from 1 through 255.

Records with defendant numbers 1 through 255 and sequence number 0 contain defendant names and other identification data, the current status of the defendant and a list of scheduled dates pertaining to that defendant. Following each of these records is a series of case history records with sequence numbers 1 through 255. Each record provides for the posting of 10 8-byte event blocks showing the event, date of the event, changes in defendant status or location, and the location of any pertinent related data, such as a scheduled date, or a related case number. When an 88-byte record is filled, and a new transaction is to be posted, a new increment of 88-bytes is added to the case history, and the next available sequence number is assigned.

Indexes to the master file are maintained by docket number and by case status. Other indexes can be implemented as required. When records must be added, they are first added to an overflow file, and an overflow index maintains linkage with the master file. As required, the overflow file can be merged with the master file, and the whole file reindexed.

MODIFYING THE SYSTEM:

To implement the COURTRAN processor in a new environment, the worst case involves the construction of 5 external tables and an attorney file, and possibly the writing of 3 or 4 subroutines to replace certain low-level clerical modules in the system. This worst case assumes a different case classification structure, and obviously and universally, a different list of judges and attorneys. In the best case, only the list of judges and attorneys need be constructed, but the lack of standardization from court to court makes the worst case most likely.

The mnemonic dictionary and the attorney lists can be implemented easily by card input to existing programs. At this writing, the judge list must be constructed manually, but it will soon be handled by card input.

Although the physical implementation of a new mnemonic structure is simple, it should be noted that the implementation depends on the careful definition of a matrix of events. On the surface, this appears almost trivial, but experience has shown that the User is not likely to be able to supply the necessary definition without considerable thought about the details of his procedures and the type of information he needs.

Implementation of the report generator and report writer for a new environment might well involve more programming effort than the implementation of the preprocessor. If existing report formats are acceptable, with changes only in page heading and text associated with mnemonics, practically no effort beyond that of implementing the preprocessor is required. However, if different criteria are to be used for selection of cases for reports, and the reports are to be formatted differently, some new subroutines will be needed, and a new formatting and printing program will be required for each report. Even these changes are confined to well-defined minor parts of the system, and the major part can continue unchanged.

EQUIPMENT:

The system is at present operating on a 64K IBM 360 Model 30. Required peripheral equipment is: three 2311 disk drives (or one 2314), one 9-track tape drive, one 1403 printer, and one 2540 card reader. One of the disk packs contains all object modules necessary to implement the system, while 360 assembly language is used for all programs.

COURTRAN CIVIL

Most of the examples cited in the previous sections have referred to the criminal case version of the system which was developed before detailed specifications were prepared for the civil case version. COURTRAN civil uses approximately 95% of the preprocessor, master file edit and update, and report generation software, but new report writers had to be prepared and several new modules were required for those characteristics of civil case information which are unique in comparison to criminal case information.

The same file structure is used but the contents are redefined via a revised internal control table. There is an expandable *case* record — just as in the criminal version — and there are up to 256 expandable *party* records for plaintiffs, defendants and third parties. In criminal cases, one party — the

United States — is fixed, so the only party records are for defendants. Also, there are no subsidiary actions between various combinations of parties. In contrast, civil cases can involve multiple plaintiffs' filing an action against multiple defendants who, in turn, may join multiple third parties. Then, combinations of these parties may commence sub-actions in the form of cross-claims and counter-claims.

CIVIL CASE TRANSITION MATRIX:

The use of the transition matrix technique to define case flow and information needs is illustrated by the matrix established by the Illinois Northern District Court. The matrices for criminal cases usually reflect the exact event flow pattern. Illinois Northern uses what can be called a pragmatic structure. This structure follows the principles taught at the Federal Judicial Center seminars for newly appointed district judges. Emphasis is placed on the use of pretrial procedures and on the need to establish schedules for case management. The latter is sometimes paraphrased as "always have the case on calendar"; i.e., always have a date for the completion of discovery, the preparation of a document, or for a status call, hearing or trial. In accordance with these principles, the first two columns of the civil matrix distinguish between the period initiated by the filing of the case and the time at which the case is set on a calendar.

In the first stage or column of the matrix (See appendix for the complete matrix.) the status of the case, as well as the status of all parties, may continue to be updated. The events which trigger a transition from column 1 to column 2 are all scheduling events which set a date certain for an identified step in the judicial process. The reports to be produced by the system (See appendix for samples.) reflect the columns in the transition matrix. For example, one report consists of all cases which are in the pretrial stage but do not yet have a scheduled event. This report also contains information relating to the date of filing, dates on which process was served, or unsuccessfully attempted to be served, dates on which answers to the original complaint were filed, etc. This report is used by courtroom deputies to determine the existing posture of a case so they can advise the judge when the case is at issue and ready for scheduling a positive court action.

After scheduling, the case moves to the second stage and a separate report is produced which

keeps track of all pretrial activity, scheduled pretrial events, and scheduling of the trial itself. The civil case matrix also has columns for the trial stage, dismissals, and judgments. In addition, there is a matrix column reserved for "limbo" time.

NEW MODULES:

We have already mentioned the multiplicity of parties and actions in civil cases. In civil cases, a single input transaction can affect multiple party records. New modules were required for these transactions. In the civil file structure a four-character party identifier is used. The first character identifies the generic class (plaintiff, defendant, third party). The second through fourth characters identify the specific party. It was recognized that in many cases events would occur which would require the updating of all the records of active parties of a particular generic classification, and that in such cases the data input task could easily become a burden. To resolve this difficulty, the sequence number zero input record is reserved for those transactions which are to be posted to the records of all active parties described by the generic party identifier. Thus, a transaction with a party identification of "Letter 'P' Number 'Zero' " would cause the event described by the subsequent mnemonic to be posted to the individual party record of every active plaintiff in the case. The civil system automatically keeps track of which parties are currently active, *i.e.*, not previously dismissed, so that subsequent analysis of the data on any one party will show only those transactions which occurred while he was an active participant in the case.

New modules were also developed to facilitate case monitoring and planning by courtroom deputies. This involved selecting sixteen key events for inclusion in a special case status report. This particular report tracks each next scheduled key event and the "last action" which occurred among these events. Each of the scheduled or "open" mnemonics is "closed" by designated mnemonics. For each scheduled key event, there is at least one and as many as five subsequent events which provide closure. The report then shows that the planned action has occurred or that another type of action which forecloses the need for action has occurred. Basically, the new modules required to track data for this report provide special handling which meets specific courtroom deputy needs and allows replacement of considerable manual activity on his part. The only difference between process-

ing mnemonics for these key events and other mnemonics is the inclusion of a special flag in the master file record so the report will contain only the desired data. Some examples of key events are "DYS" for "discovery completion date set", or "PTS" for "pretrial date set", or "TA" for "trial taken under advisement", etc. Key events also appear on exception reports if their occurrence or rescheduling is not reported to the system by the predefined "closing" mnemonics.

Another aspect of the civil version which is helpful to courtroom deputies and clerk's office personnel and which also assists in effective report production is the capability to update two separate case records with a single input transaction. For example, when case B is identified as being related to case A which is already pending in the court, or as being involved in a multi-district litigation proceeding in which case A is the mother case, the entry of this fact via mnemonic in updating case B will also cause the docket number of case B to be automatically inserted in the case record of case A. When reports are produced all cases which are associated with case A are printed-out immediately below it. This can be seen by referring to civil report A in the appendix where case docket number 71C1111 has the docket numbers of all its related (daughter) cases printed immediately below.

INFORMATION PRODUCTS

Several years ago, enthusiasts were wont to claim the use of computers would revolutionize the administration of justice and make the courts models of efficiency. There was discussion about automatic calendaring systems and there were visions of the computer performing the administrative tasks now performed by judges and court administrators.

Needless to say, most of these expectations were unrealistic and were manifestations of the mythical and magical aura which often surrounds discussions about computers. A realistic appraisal reveals that both designers and users are still working in virgin territory in learning the ways in which computers can be most useful as a tool for court administration. But it is becoming clear that computers, used properly, can serve as a tool to make supporting operations more effective, can improve the types of information available for managing the business of the courts, and can provide a capability for evaluating the effects of procedures and identifying the composition — and sometimes the causes — of delay. Thus the product of this tool is usually

information in the form of printed notices, orders, subpoenas, etc. or in the form of reports. The following sections describe some of the information products which have been developed or are being developed for pilot operation of the COURTRAN system. (See appendix for samples.)

INDEXES AND INVENTORIES:

Each court maintains indexes which provide organized access to information about both pending and closed cases. COURTRAN replaces the manual effort required to produce these, adds several which it was not feasible to prepare manually, and includes additional information not previously provided. Some of these are:

1. *Party Name Indexes* -- Alphabetical indexes are prepared for both civil and criminal cases. For criminal cases, this includes the type of offense, the judge, prosecutor and defense counsel and the age of the case in days. Indexes for civil cases include the type of case, the docket number, the judge and attorneys' names and the age of the case. Those indexes which include only pending cases can be considered as an inventory. Indexes which include all cases filed, both pending and terminated, can be used as permanent party name indexes.
2. *Docket Number Indexes* -- These are usually produced for pending cases only. In the past, courts have prepared consolidated listings twice a year and have added monthly supplements. Now, a consolidated index can be produced monthly. This eliminates the manual effort and makes the index more usable since it is not necessary to refer to several supplements.
3. *Judge Indexes* -- Since most federal courts use the individual assignment calendar system, each judge has a number of cases for which he is responsible. Indexes are prepared of each judge's cases in both party name and docket number sequence for inventory purposes. These also contain other items of information as noted above. The system also keeps track of transfers among judges and transfers to and from the Calendar or Executive Committee.
4. *Attorney Indexes* -- These indexes are in alphabetical sequence by attorneys' names and list all the cases for which they are the attorney of record. In the criminal version of COURTRAN, a special prosecutor's report is prepared which lists all cases for which a given Assistant U.S. Attorney is responsible and groups these according to the judge to whom the case has been assigned. These reports have been produced for approximately one year. Indexes for defense counsel are currently being implemented.
5. *Special Indexes* -- Other indexes include: an alphabetic list of all fugitives in criminal cases; all defendants currently committed; and all defendants awaiting sentence. Additional indexes can be provided according to any categorization required by the court. The user can specify the elements of information which he wants to appear on the index.

CASE STATUS REPORTS:

Case status reports are prepared for both civil and criminal cases for each judge, and for the total court. These reports are organized according to the stage and event categorization established by the

transition matrix which is prepared by each court prior to implementation. Status reports include not only the last action which occurred on a case but the next scheduled dates for each case, whether this be a trial, a hearing or a due date for a report.

MONITORING REPORTS:

This type of report is used by clerk's office staff and courtroom deputies to monitor scheduled events and to assure that all necessary steps have been taken by supporting operations to provide for the occurrence of the event. Included in this type of report are "Exception Reports" which list cases for which some planned event has not occurred within the prescribed time period or on the scheduled date.

Appellate events are also included in monitoring reports. Reports on this stage start with the Notice of Appeal date, the date the record is due in an appellate court, the date a transcript is ordered from the court reporter, the date the finished transcript is filed with the district court clerk, and the date the record is transmitted to the appellate court. The appellate decision is another input which triggers notice on any district court action required.

STATISTICS:

The system also produces statistical reports showing filings and terminations (by type of disposition) for each judge and for the total court. These reports are automatic by-products of the system and replace the manual effort previously required for their preparation.

JS REPORTS:

A case opening card is automatically punched when a criminal case is filed and case opening and closing cards are automatically prepared for civil cases in the format required for the monthly JS (Judicial Statistics) reports required by the Administrative Office of the U.S. Courts for national statistics.

SPECIAL REPORTS:

The system not only produces periodic administrative reports but has an open-ended capability to prepare special reports to evaluate case processes or to test any of a wide variety of research hypotheses. The COURTRAN report generator makes it possible to retrieve any combination of event mnemonics and information on case characteristics

from the data which exists in the master file. For example, one might wish to know the difference between criminal cases in which omnibus hearings were held as compared to cases where this technique was not utilized. In this type of analysis, we look for the time between filing and trial for those cases going to trial, the time between filing and guilty plea for those cases following this route, the number and type of motion hearings, etc. The report is then in two parts, the first part consisting of data covering cases which did not involve omnibus hearings and the second part involving the same type of data for cases which involved omnibus hearings. The COURTRAN report generator produces means, medians and ranges for each time interval specified for any report. The same data can be produced in a separate part of a report with all of the non-trial and "limbo" times excluded in order to show the actual elapsed time during which the case was under the control of the court.

Using the report generator, a special report writer has been prepared which produces time profiles for cases following six different paths through the court system. These reports are modeled after those prepared for the speedy trial research project conducted by the Center last year. In courts where COURTRAN is operating, these reports can be produced as a system by-project. For these time profile reports, cases are separated into six categories, *vis.*, (1) those which resulted in trial, conviction and sentencing; (2) those which resulted in a guilty plea after a trial had begun and were subsequently sentenced; (3) those in which there was a guilty plea before trial followed by sentencing; (4) those which were tried and acquitted; (5) those which were dismissed before trial and (6) those dismissed during trial. Each of these is broken into stages which consist of the time between filing and the end of pretrial, the time between the beginning of trial and the verdict or judgment, and the time between a conviction or guilty plea and sentencing.

Time profile reports are an excellent example of the procedures used by the Center in converting research techniques into operating system capabilities. As a result of close coordination and planning, the COURTRAN system will be used by the Research Department for a similar analysis of civil cases using data collected during the summer of 1972. The COURTRAN Civil Research version will be used for this analysis. A somewhat similar report can be used for analyzing the pending case-load inventory of a court. This particular report segregates cases by type of offense and shows the age in days for each case in each category along

with the mean, median and range for all cases in a given category.

Because of the flexibility of the COURTRAN software, it is possible to prepare almost any type of analysis which can be defined in terms of the case characteristics and events which exist in the file. In the civil speedy trial research project for example, we will be analyzing the nature of motion practice (types of motions, number of motions, number of hearings, etc.), the relationship between the time of setting a trial date and the total time for disposition of a case and differences between the paths followed by different types of cases.

FUTURE APPLICATIONS

Once the system versions described herein are all implemented we will be studying possible applications for magistrates' courts and appellate courts. The present criminal version starts with the filing of an indictment or information, which is the point at which a case is assigned a district court docket number. Although many federal criminal cases are "Grand Jury Originals," and therefore start at this stage, a complete information system should include events handled by U.S. Magistrates for cases where arrest, initial appearance and bail decisions occur prior to a preliminary examination and/or a binding over to the grand jury. This part of case flow will be handled by preparing a separate matrix and including it in the mnemonic dictionary. In addition, a docket number linkage will have to be included since a case is given one docket number by magistrates and is given a new docket number if and when an indictment is returned.

A version of COURTRAN for appellate courts will be relatively simple compared to the civil and criminal district court versions. An initial matrix, which includes only major events is shown in the appendix. Although initial design studies have been conducted in one appellate court, implementation of an appellate version of COURTRAN will not be seriously considered until sometime in 1973. In the meantime, the initial studies show that the present software will be directly applicable for an appellate court information system.

During the coming year, the report cycle period will be reduced to weekly and possibly daily processing. At this point in time reports are being produced once a month. Plans are also being made to link criminal and civil data for several report needs identified by the pilot courts.

During the experimental operation period, we are finding that a design which facilitates evolu-

tionary growth is paying off. Every time a new information product is introduced, it generates new insights into other system uses and the need for changes in existing procedures. Both court personnel and members of the Center staff are continuing to learn as a result of this process. Although the system was designed to be simple to operate, we were still concerned with the problems an operator in a local court might have when he was "on his own". However, we have found that software complexity has resulted in ease of operation and although there was a short debugging period, no major problems have arisen since operation began.

We can conclude at this time that the system is effective. We cannot claim any dramatic changes in the administration of justice as a result of its implementation, but there are numerous instances where the objectives of court management were facilitated by the use of the system and a significant amount of manual effort has been eliminated thus

freeing-up time for more important parajudicial duties. The increasing trend of usage indicates the pilot courts are integrating the system more and more with their procedures. Our major worry at this point is that the demand will increase faster than we can provide the necessary technical services.

FOOTNOTES

- 1 One example which may be helpful to the reader is motion activity. A general level would consist of the fact that a motion hearing occurred. Another level would include information on the type of motion. A more detailed level would consist of the dates when each motion and reply was filed, the date of the hearing, if any, the date of the decision, and the decision by the court on the motion.
- 2 It should be emphasized that a national system has to use judgeships as a dividend in computing caseload per judge. Adjustments based on local situations would unduly complicate the statistical normalizing process. Also data for determination of the need for additional judgeships cannot be based on the actual number of active judges.
- 3 A transition is denoted with the row to column numbers. The row numbers represent the same stages listed above the columns.

END