TRANSPORTING PROGRAMMING SYSTEMS

David J. Terwey
Deputy Head, Data Processing Division
International Fertility Research Program
Research Triangle Park, North Carolina 27706 USA

During the past four years, the International Fertility Research Program (IFRP) has designed, implemented, and transported information management and analysis systems from a home computer center in the United States to host centers in Asia. The problems we encountered in transporting these systems can be categorized as logistical, operational, and technical. Technical problems directly related to converting the programming systems are the ones most often addressed, but in our experience, prove to be the easiest to solve. Operational problems such as availability of computer time and system utility programs required some planning, but mostly patience to overcome. The most difficult problems that we faced were selecting a target facility, allocating sufficient resources to prepare the conversion team, and other similiar logistical problems inherent in preparing to transport any large scale programming system. Close coordination and communication with personnel and program users at the target center proved necessary to solve many of these problems.

INTRODUCTION

The International Fertility Research Program (IFRP) began as a contractual research project within the Carolina Population Center at the University of North Carolina, Chapel Hill in July 1971 and became an independent research institution in February 1975. Its major purpose is to analyze the effectiveness and the acceptability of new methods of fertility control and to report these findings rapidly. An extensive network of reporting clinics and hospitals located in more than 35 countries has been developed over the years. Because the IFRP analyzes large quantities of data rapidly, the program's need for computerized information management and analysis systems was recognized at its inception.

The IFRP conducts studies in six areas of fertility control and contraception: Pregnancy Termination, Menstrual Regulation, Female Sterilization, Intrauterine Devices, Male Sterilization, and Systemic Contraceptives. For all studies within each area, there is a consistent core of data items, but few non-demographic data items are common to all study areas. The first major design decision our Data Processing Division faced was whether to have separate information management and analysis systems for each study area or to have a single system flexible enough to handle all areas. Since it was extremely important to develop systems rapidly and since the data collection instruments were to be designed sequentially, we chose to develop separate systems for each study area.

After making several false starts and implementing some stop-gap systems to handle immediate problems, we began in early 1972 to develop the general information management system and the analysis system for each study area. The two main goals of the information management systems are:

---

*This work was supported in part by grants from the International Fertility Research Program, Research Triangle Park, North Carolina (AID/csd 2979).

- To strenuously verify the consistency of data between items for each case. When apparent discrepancies are found, the system provides an efficient method of querying the contributting clinic or physician to verify or correct the items in question.

- To provide an accurate overview of the amount and "quality" of the data contained in the data base for both the IFRP's Administrative and its Design and Analysis (research) Divisions.

The IFRP analysis systems have two major goals:

- To be easy for nonprogrammers to use in two ways. First, the systems are easy to invoke so that setting up general-purpose runs requires a minimum of researcher's time. Second, the systems are clearly labelled so that the researcher can use the tabular output without having to reformat or relabel it.

- To be general enough to handle most, if not all, of the routine analysis requests. Separate special analysis programs and setups of statistical packages are available for extremely detailed or unusual requests or to further investigate trends or points of interest found in examining the general analysis.

From the beginning we designed our system to meet these goals by providing for:

- Flexibility. The nature of IFRP research required that the systems be sensitive to differences between studies in the same study areas, but more important, that they react quickly to new developments in the study area itself.

- Efficiency. As with any computer project, it was very important that the programs could be both implemented and executed rapidly and economically.

- Transportability. In the early design of our system, we recognized that regional processing centers would eventually be required to handle some of the anticipated flow of data.

Designing the system for transportability was essential to the success we enjoyed in converting our programming systems to other computer facilities.

While designing and implementing our systems, we made several decisions with the ultimate transporting of the systems in mind. First, all programs were written to run in 100K bytes or less under IBM OS/MVT, even though they were developed on the 3 megabyte 370/165 at the Triangle Universities Computation Center (TUCC) in Research Triangle Park, North Carolina. Many of the programs required a rather substantial overlay structure to accomplish this, but the conscious effort to conserve memory undoubtedly improved our programs. Second, we wrote all new programs in Fortran to run under Fortran G and OS/MVT and used some existing utility programs written in 370 Assembler, intending eventually to rewrite them in Fortran. We felt that these two factors would allow us to run our systems on many medium to large computer systems with minimal conversion and would also improve efficiency.

TRANSPORTING PROGRAMMING SYSTEMS IN DELHI, INDIA

The first test of our planning for transporting the systems came much sooner than we would have liked. Early in 1973, many Indian contributors joined together to form the India Fertility Research Programme. As soon as it was logistically feasible, the DP Division of the IFRP began to prepare to transfer our existing management and analysis systems to a computer facility in India.

After the site of installation had been identified to be the IBM 360 model 44 at the Delhi University Computer Center, we began to implement our systems from October to December 1973. I made a preliminary investigatory trip and, with Mr. Robert Taylor, then a programmer for the IFRP, a second trip to work on converting the systems. Even though goals of this effort were limited, we were able to successfully implement several IFRP programming systems. Mr. M.C. Khandekar, who was acting director of the Delhi center at that time, M.N. Gupta, and other members of the staff at the computer center, and Mr. Kalyandrug Sivaram of the Rural Health Research Center at Narangwal were all very helpful in the implementation. The India Fertility Research Programme used these programs as a stop-gap measure until we could deliver a more complete set of systems fourteen months later. During February 1975, Mr. Sam Gilbert of the IFRP worked with me in Delhi on the second installation--this time with an expanded and updated set of programs to handle four of the six major study areas. The people listed above and Mr. Raj Bhatia again were instrumental in the success of the effort. Since then, Mr. Bhatia, among others, has been using these programs to manage and analyse data from a network of contributors sent to the India Fertility Research Programme in Calcutta.

Since the second conversion in 1975, the systems at IFRP continue to be refined and expanded. A third trip will probably be made in the fall or winter of 1976 to update existing programs and to add several new management and analysis systems.

In addition to the conversion in India, we have begun preliminary planning to establish regional processing centers with varying degrees of computerization in several other areas of the world. The South East Asia Fertility Research Program in Kuala Lumpur will probably develop such a center within the next 18 to 24 months; preliminary investigations for a suitable computing site have been underway for some time. Programs are also underway in the Sudan and Columbia to develop centers during this time. The IFRP is extremely interested in identifying and overcoming the problems of transferring computer programming systems to countries in South East Asia as well as other parts of the world. From our experience, we have identified three types of problems that may occur in other conversions: logistical problems with personnel, supplies, and equipment; operational problems with computer time, systems tools, and maintenance; and technical problems with the systems being transferred.

LOGISTICAL PROBLEMS

The logistical problems of conversion proved to be the most frustrating problems we faced. They began well before the conversion team was even selected. The first and most crucial problem is to find a site where the system can be implemented. Personnel in the country or region where the systems will be transferred may spend months looking for a suitable geographic location and computing facility. The supplier of the programming system must establish a detailed communication with someone familiar with the target center. From this liason for the target center, the supplier will require information on such major operational factors as computer and peripheral hardware, available operating systems, compilers, assemblers, linkage editors, and utilities. The suppliers will also need to know about normal operating procedures for shared computer time or block time at the target facilities. If several operating systems are available, procedures for using the desired system and the percentage of time it is normally used must be considered. The cost of computer time--which can vary greatly within an area--should be compared with that at other available sites. Once procedures, available time, and costs at all available sites have been established, it is fairly easy to limit the number of suitable facilities by careful communications between the software supplier and the personnel in the target country. But no matter how detailed and promising the information seems on paper, firsthand use of the facility is necessary to finally decide if a computing facility is acceptable. It is worthwhile for host personnel to use the facility but finally most preferable for a member, if not the leader, of the conversion team to use it. Our conversion team's firsthand experience saved considerable time and resources in preparing the systems for

conversion. The experience can also help the conversion team establish rapport with both the personnel who will eventually use the packages and the staff at the facility that will supply the computer resources.

While the computing facility is being identified, members of the conversion team should be selected. We feel that any project that justifies sending one representative to a target site justifies sending at least two. There are many advantages to sending more than one person. If the conversion task is segmentable, division of labor is reason enough. If any member of the team encounters significant technical or programming problems, he can usually solve them more easily in discussions with colleagues. Establishing such a relationship with people who have not worked together is quite difficult. If one member of the team gets sick or cannot function for some other reason, having a backup person is desirable. Even if there are no problems and no one gets sick, the presence of a coworker and friend in an unfamiliar setting and society is probably worth the additional cost in time and resources of sending two people on such a venture.

Members of the conversion team should be selected with an eye to both their technical ability and personalities. Even though our programs are clearly "application" programs, the technical problems we encountered in implementing them and the solutions to these problems required that the team members thoroughly understand the hardware and operating systems. The reason for this will be discussed later. Almost as necessary as these technical qualifications, team members should also be selected who are able to work in new surroundings and react efficiently to unexpected problems.

The task of freeing up the time and other resources necessary for the conversion team to adequately prepare for the actual effort proved to be a logistical problem in itself. Unless the supplier is in the business of providing software to other installations, preparing a programming system for conversion consumes a considerable amount of additional time. Since every conversion is different, the specifics of preparation will vary greatly, but certain aspects of our preparation can be generalized. First, any known program changes should be made at the home site, if at all possible. Second, a complete dry run of the systems will help detect any bugs created by the changes or any sections of code, JCL, input parameters, or other information necessary for system operation. Third, such standard practices as making multiple copies of the tapes or other media used for transportation and making complete listings of the source being taken are worth the extra preparation, even if their only use is to reduce anxiety over the possible destruction of one's only copy. Regardless of the specifics of a conversion effort, freeing up the time to adequately address these three factors is a major problem for most organizations.

The lack of technical supplies such as disk packs, tapes, and multi-part paper in the target countries can create another logistical problem if the supplier has not planned carefully. We found this problem to be particularly acute in India. Such supplies are readily available in the United States and, because of competition from independent manufacturers and lack of customs duty, are generally much less expensive than in South East Asia. But trying to plan ahead and bring such items from home can lead to other problems ranging from payment for excess baggage to lengthy discussions with customs officials about why one needs to bring these fourteen boxes into the country. The problem with customs officials is particularly noteworthy in India, but exists in other countries in this area as well. And it must be remembered that the problem may exist in any country entered enroute to the target country. These officials are often less familiar with computer paraphernalia than officials in the U.S. and Western Europe. Explaining the purpose and value of two 2316 disk packs at 5 a.m. to an Indian customs agent is extremely exasparating and, in our case, resulted in their being stored at the airport for our entire visit. The usual logistical problems of travel such as language, food, and accomodations prove trivial compared with these problems of supplies.

OPERATIONAL PROBLEMS

Operational problems did not require as much time or resources to overcome as the logistical ones, but they did require more patience. Our major operational problem at Delhi was computer time. Even though Mr. Khandekar, the acting director, and Dr. Grover, the operations manager at the Delhi University Centre, scheduled time for us generously, we were spoiled by the time we had had at home. This is probably true of most software suppliers in the U.S. When computer time is more available and economical than programmer time, it is very easy to use computer time extravagantly for programming and debugging. In Delhi and probably throughout South East Asia, the limited and valuable computer time must be used prudently. Our conversion team had to consciously adjust programming practices to increase the amount of desk-checking and hand preparation of proposed program changes. We gradually learned not to waste what time was available pursuing possible fixes.

System tools and utilities that we took for granted at home were not available. Interactive program editors, short-cut utilities for maintenance jobs such as moving datasets and backing up and mapping disk packs are usually available at the larger computing centers in the U.S. Again in Delhi, some of these tools were not available; others were available through more cumbersome but equivalent IBM-supplied utilities. We had to plan to bring from home some of the tools we felt absolutely necessary. A secondary benefit of this planning was that we added to the program library of the target facility, a positive feature in itself.

Several secondary operational problems required more patience than anything else. These ranged from power and air conditioning failures that could wipe out an entire day of time for us, to maintenance problems such as unavailable replacement parts which could keep the entire computer site down for some time. Such problems are unavoidable in any location but seemed more prevalent in India than we were used to or expected.

TECHNICAL PROBLEMS

Finally, we faced technical problems directly related to converting the programming systems. When systems are being transported, technical problems have usually been most thoroughly prepared for; however, our experience suggests that the technical problems that do arise are by far the easiest to solve if all problem areas have been equally considered and prepared for. It is much easier for a good conversion team to solve programming problems at a suitable computing facility than it is for an underqualified conversion team to install an error-free system at an unsuitable computing facility with insufficient tools and preparation.

In any case, both the predictable and unpredictable technical problems that we faced are worth describing, if for no other reason than to possibly help other, similar teams plan conversions. From our advance work, we were able to predict two types of problems, but even so, all their ramifications did not surface until the actual conversion. The first of these predictable problems arose because the target facility was using a slightly different operating system, even though both host and target systems used IBM OS (one used MVT, the other PCP). The differences were small and varied but required a substantial amount of reprogramming which of course produced a small but predictable number of bugs. Also, our hosts had much less experience--both from an operational and systems programming standpoint--than we did with IBM OS so that we had to rely on our own experience in unusual situations.

The second predictable problem was that certain subprograms and routines considered "standard" on our home system were not available. Most computing facilities write or adapt special subprograms which are commonly used at that installation and call them "standard". It is easy to take the existence of such resources for granted. Our team had considered this problem and brought along "all" routines we

expected to need--only to discover that the routine we used to return the current date in the format that we expected was not "standard" after all.

The "unpredictable" problems arise because no operating system or control program of any large computer system is error-free. It is only reasonable that the errors that affect a specific facility's production will be attacked by that facility first. If there are enough errors, as is the case with IBM OS and Burroughs MCP, certain errors will always remain lurking in the recesses of the control program. Since different programmers use different techniques, the team should expect to unearth such errors during conversion and should watch for them. However, considerable experience and expertise in the so-called "systems" area are necessary to accurately isolate and identify a problem as a system error rather than a conversion error.

A second source of surprises was the differences in standard operating procedures. In the absence of specific requests or information from us, operations personnel at the host site would follow their usual procedures which sometimes differed from ours. We had several unexpected problems involving public and private disk packs. First, the default size for the Volume Table of Contents (VTOC) was one track. Since we requested no special disposition, our packs were so initialized. Because we use many datasets, this caused a substantial amount of duplicated effort when we ran out of directory space and had to rebuild the VTOC as well as most of the datasets we had created. Second, our hosts scratched all nonsystem datasets on public disk space every day. Because we assumed the semi-permanancy of such files, as is the case on our home system, we lost several datasets and then had to rebuild them. There was, of course, nothing wrong with our host's procedures: we should have thought to ask about them. The conversion team can expect to have to contend with such unpredictable problems because it is impossible to learn all of any facility's standard procedures in a short time.

CONCLUSION

In summary, we learned that the successful transporting of programming systems requires advance planning and close coordination and communication with personnel and the ultimate program users at the host facility. If the system is designed with transporting it in mind, many if not most of the technical problems will be solved before they arise. In any case, planning for a conversion is at least as difficult as actually implementing the converted systems. The target site and the conversion team must be carefully selected. The conversion team should have as much diverse experience as possible, especially in the "systems" area. The conversion plan should include a preliminary visit to the site and dry runs at home and at the site if possible. The dry run at home is essential. Arrangements should be made for all supplies and programming tools necessary to be available. If there is any doubt that any supply item, manual, utility program, or subroutine will be readily available, it should be brought from home. Transporting this much material can be a major problem in itself and should be considered in the planning.

Secondly, we learned that close coordination and detailed communication with personnel at the target site as well as with the people who will actually use the programs in the target country is an absolute necessity. With effective communication, the conversion team can be warned of impending problems and shortages as well as the strong points of the target system. Such communication will result in a more successful and enjoyable conversion than any team working independently can accomplish.