

JAMES F. COREY
Systems Librarian
University of Illinois
Urbana-Champaign, Illinois

Negotiating Computer Services within an Organization

THE OTHER PAPERS PRESENTED at this conference deal with negotiations between libraries and outside organizations, where the end product of negotiation is a written contract of some sort. This paper, by contrast, attempts to describe the possibilities for negotiation when the library uses the computer services of its parent organization.

Libraries have frequently had difficult relationships with their organizations' computer centers. For example, in 1962 the University of California at San Diego (UCSD) Library developed and commenced operating one of the country's first serial control systems.¹ The system was gradually improved and was running steadily when, in 1967, the Control Data Corporation computer at the university computer center was replaced with an incompatible one made by RCA. The library, in a frantic effort, reprogrammed its serials control system in time to have it running on the RCA computer. Just two years later, in 1969, the UCSD computer center changed from the RCA machine to a Burroughs computer. The library was again forced to reprogram. According to one account from UCSD: "The situation concerning lack of stability and scheduling of computers . . . contributed significantly to development costs, operational costs through conversion requirements, and . . . generally slowed progress."²

Another example can be taken from the University of Illinois at Urbana-Champaign where, in July 1973, the director of the university

Office of Administrative Data Processing sent a letter to the director of the library on the subject of a computer circulation system, promising that the Office of Administrative Data Processing "will commit to a May 30, 1974, deadline for delivery of a system to the library staff."³ The system was delivered in May 1976 — nearly two years late. Unforeseen circumstances outside the control of the Office of Administrative Data Processing were in part responsible for the delay at Illinois, and the UCSD computer center may have had compelling reasons to switch computers twice. But an important question remains: What can the library do to protect itself from poor and costly service from its organization's computer center?

A library negotiating for computer services within its own organization cannot have a "contract" for services in the legal sense. Legal contracts can only be established between separate organizations. The final resolution of a contract dispute is court action. If two units of the same organization were to enter into a "contract" with one another, and subsequently the terms of the contract were not met, the organization would be faced with the situation of going to court to sue itself. Thus, libraries which use their organization's computer centers must find a substitute for the legal contract. Many libraries have embarked on projects with very little prior understanding of how the work was to be done and a vague feeling that problems would be solved as the work progressed. Some of these projects probably went smoothly. In others, issues may have been settled reasonably and amicably as they arose, with no detriment to the library. In other cases, however, serious problems based on misunderstanding have left both the library and computer center frustrated and angry. In the absence of a clearly written prior agreement, small projects are more likely to be completed than large ones, simply because there are fewer details which are potential sources for misunderstanding. The larger or more complex the project, the more important a clear understanding becomes.

Since a contract cannot be written between the library and computer center, what can be done? The answer is to draw up a written document that is like a contract in every way except in legal authority. Since the document is not a contract, it can be called an "agreement" or "joint memorandum of understanding." There are two reasons why a written agreement drawn up in advance of a project is valuable. First, the agreement will clarify what is needed for successful development and continuing support of a system. Better planning, especially development planning, will result and implementation schedules will be projected with more accuracy. Secondly, when problems arise, the agreement document can be used as persuasive leverage to obtain compliance from the other side.

Organizations which maintain a computer center as an internal unit of the organization may have one of two kinds of policies with respect to the degree of centralization of data processing resources. Some organizations permit their departments to employ analysts and programmers, but the computer equipment is centralized and shared by all the departments. In this case, the library will need only to negotiate for the use of the computer. Other organizations have a centralized pool of analysts and programmers as well as a centralized computer facility. Analysts and programmers are assigned by the computer center to departments on the basis of department needs, overall organizational priorities and, in some cases, departmental ability to pay for the services. In this latter case, the library must negotiate the services of analysts and programmers in addition to computer usage.

To cover all the detailed points that could go into a comprehensive agreement would require a document several times longer than this paper, but the general areas that should be covered by an agreement can be outlined and some commentary can be provided on the importance of each area. A suggested outline of major areas for negotiation is given in Table 1. Points on the outline will vary in importance depending on the nature of the application and whether the system is batch or on-line. Many points are interrelated. To reinforce the importance of reaching prior agreement, several examples taken from actual situations will be described to illustrate what can happen when issues are not agreed upon. In many of the examples, the library is not identified in order to avoid embarrassment to institutions with which the author has been associated.

Agreement for Machine Services

The hardware required to support the library is one of the first points to negotiate. The amount of main memory needed for the library's application should be discussed. While it is true that computers are becoming sophisticated to the point that main memory assigned to a program can be dynamically adjusted depending on the immediate demands placed on the computer, the library should nonetheless attempt to determine whether there are any restrictions on the amount of main memory that will be available.

Secondary memory, i.e., disks and tapes, can be a potential negotiating problem. Since library files are large, the library must negotiate forcefully, especially for disk space; otherwise the library may have to make serious compromises. One academic library was forced to use truncated titles in its circulation system and ended up with such titles as *A Priest for Ever: A Stud* (short for *A Study of Typology and Eschatology in Hebrews*) and *The People of Ancient Ass* (for *Assyria*). College students chuckle when they get overdue notices for such titles; the reaction of public library users might be less favorable.

TABLE 1. MAJOR AREAS FOR NEGOTIATION

Agreements for machine services

1. hardware
 - a. main and secondary memory
 - b. communications lines
 - c. terminals and other I/O devices
 - d. character set
 - e. stand-alone equipment
 - f. maintenance
2. computer availability
 - a. hours of service
 - b. tolerable down time
3. operating system support
4. priority
 - a. job scheduling
 - b. response time
5. production
 - a. schedule
 - b. logistics (delivery of input and output)
 - c. forms and supplies
6. price
7. growth
8. long-term hardware and software continuity

Agreement for personnel services

1. method for assignment of analysts and programmers
 - a. delivery of finished product at specified time
 - b. time and materials per project
 2. change control
 3. documentation
 4. acceptance testing
 5. program maintenance
 6. program extensions
-
-

Communications lines must be specified as to type (dial-up, leased point to point, leased multipoint), speed and location of termination points. Lines must be compatible with terminals, so they should be specified at the same time. The number of hard copy terminals and the number of CRTs, along with the features to go on each, must be decided. Some terminals in key locations may require extra features, such as a tape cassette attachment to use when the computer is down. If the library needs other input/output devices such as optical scanners, the computer center must agree to support them. In some cases, the library may want the capability to add an attachment to their terminals in the future. The attachment may not work on terminals currently supported by the computer center. This could require the computer center to support a new type of terminal which otherwise would not have been selected.

The character set to be used should be planned carefully, because it may not be necessary to require the same character set on all input/output devices. Some libraries are satisfied with terminals with only upper- and lowercase and a few special characters to enter cataloging data. Foreign characters and diacritics are handled by the use of an escape character preceding one of the regular characters. However, when the cataloging data are printed on 3x5 cards, the full ALA character set is required and is mounted on the computer line printer. Other libraries want the full ALA character set on their cataloging terminals but will accept just upper- and lowercase on terminals used for searching or serials check-in.

When all of the hardware that is attached directly or remotely to the computer has been specified, the library should not overlook hardware that is separate from the computer. Optical character readers, Hollerith punch machines, bar-coded label printers, and computer output microfilm machines are just some examples of equipment that may be needed but are not attached to the central computer equipment itself.

The hardware "maintenance" specified in Table 1 refers in this instance to maintenance of computer equipment installed in the library. While the computer center obviously has responsibility to maintain equipment on its own premises, it cannot be assumed that the computer center accepts responsibility for computer equipment at the library. Now that on-line systems are becoming more prevalent, the most common pieces of equipment in libraries are terminals and modems. Libraries may also have keypunches, terminal controllers, concentrators, multiplexors and minicomputers. The equipment may be maintained by the computer center, the original manufacturer, or a third party. The University of Illinois Library has some terminals maintained by IBM, other terminals maintained by General Electric, others by the campus computer center, and still others by an independent maintenance firm located 120 miles away in Indianapolis. Some of the modems are maintained by Illinois Bell, and the rest are repaired on campus.

The requirements of the library for computer availability are quite important for on-line systems. One library discovered in the midst of developing a circulation system that its computer center was reluctant to run the system during all of the hours that the library was open. To meet library hours, the computer center would have to renegotiate its contract with the union governing computer operators, and would have to reschedule preventive maintenance and system test time. The computer center proposed that the library record circulation manually during certain hours and enter the transactions later when the computer was available. After moments of serious doubt, the library was able to convince the computer center to make the computer available during all open hours.

Down time can also be negotiated to some extent. Not all down time results from unanticipated hardware or software failure. Computers are often taken down deliberately to do system tests, switch equipment into or out of service, or perform maintenance. The library, by informing the computer center in advance of its peak periods and being willing to do without the computer during quiet periods, stands a good chance of getting a favorable agreement from the computer center for scheduled down time.

Operating system support is not normally a concern to the library. The library states its requirements in terms of functions to be performed and operational considerations such as hours of service. Occasionally, however, the operating system can be an issue. Most frequently it is an issue when a library tries to obtain programs written elsewhere. Programs written for one library may have been written to be used with a version of an operating system not in use at the other library's computer center, even though the two computer centers have computers of identical make and model. To cite two technical examples, programs written for IBM's OS won't run on IBM's DOS and programs written for VSAM won't run with ISAM. Negotiation will determine whether the operating system will be changed, the programs rewritten, or the hope of transferring the programs abandoned.

Anyone who had experience with OCLC in 1974 knows that response time is an important point that should be negotiated by the library. Response time is determined by so many complex interrelated factors that it is extremely difficult to predict in advance, even with complete knowledge of the design of the application. If response time is slow, the problem can be anywhere, e.g., file structure, indexes on a slow disk, communications line too slow, or insufficient main memory. The correction may require changes that the computer center is reluctant to make. Prior agreement on a reasonable response time is essential if a reluctant computer center is to be persuaded to make the necessary improvements.

The term "production" in Table 1 refers to the day-to-day running of a system once it becomes operational. Details concerning daily operation are frequently left until a system is near the end of the development stage. Scheduling is one such detail commonly ignored. Analysts and programmers will, of course, have an initial rough idea of the frequency of use of each program, e.g., monthly, weekly, daily, on demand, or at fiscal year end. Late attention to the specifics of program scheduling can, however, bring some unpleasant surprises. One library has a monthly accounting report that should be produced at the end of the month. It was discovered that, because the computer center already had a full schedule of month-end jobs, the library's job must run on the twenty-sixth

of the month, reflecting transactions through the twenty-fifth. This arrangement is tolerable, but does require a special adjustment run at the end of the fiscal year to include transactions for June 26-30 in the year's accounting. This adjustment run must wait until mid-July, because the computer center is also saturated with year-end jobs.

Logistics are also frequently ignored. The library has an obligation to generate input for scheduled programs on time — but who transports the data to the computer center? Does the library carry the data to the computer center? Does the computer center come after it? Similar questions must be answered for output. One library found itself in the situation where the previous day's purchase orders were delivered each morning to the library by the computer center's courier service, while the previous day's overdue notices were not. The circulation system was developed two years after the book-order system. When the circulation system became operational, the courier service was declared to be fully loaded. After several months of negotiation, it was finally decided that the 6'2", 210-pound courier who delivered to the library could carry the extra two pounds of overdue notices — provided that he could leave them at the acquisitions department. The circulation department is delighted to send a 95-pound weakling down the hall each morning to get the overdue notices from the acquisitions department, presumably saving the weary courier from complete exhaustion. Another library in the Pacific Northwest lost their delivery battle. During the rainy winter season the library prints purchase orders only once a week because of the nuisance of trudging through the rain to the computer center to pick them up.

In the areas of price, growth, and hardware and software continuity, the library is essentially negotiating for the future. There have been many instances where libraries have been attracted to computer centers by offers of either free or unbelievably low prices. The offers were made when the computer centers had excess capacity. Invariably, computer usage continued to increase, excess capacity vanished, and the library was told to pay the standard rate, which amounted to an enormous increase.

The library should also attempt to elicit a commitment for future growth. This is especially important when the computer center has purchased its computer. One large university library developed a circulation system to be installed initially in its heavily used undergraduate library. After a year of operation, the library wanted to extend computerized circulation to all of its branches. It found to its dismay that, during the one year of operation, other university departments had developed numerous applications. The computer, a purchased machine, was saturated. The university could not afford just then to buy another computer; consequently, the library could not expand its circulation system.

Long-term hardware and software continuity is one of the most difficult points to negotiate. Computer center directors like to upgrade or improve their facilities. The impact on the user is often given little attention, especially in organizations where programming is not centralized within the computer center. Where programming is centralized, the computer center will have the responsibility to modify programs to run on new equipment. The library need not negotiate long-term continuity. However, libraries which do their own programming should seriously negotiate continuity. The experience of the University of California at San Diego should be enough to prove the point.

The library must not only be thorough in negotiating each of the above points individually, it must also cover interrelationships between points. For example, jobs usually can be assigned more main memory at a cheaper price at night than during the day. Terminals must be compatible with communications lines, and both must be supported by the teleprocessing software that is supplied with the operating system. While that much surely seems obvious, one library forgot to relate terminals, lines and teleprocessing software to job scheduling. The library bought terminals to be used for a technical services data collection application. Library staff keyed transactions into the terminals' local memory during the day. At the end of each day, the data stored in the terminals were to be sent down a phone line to a large computer. The terminals were installed, the proper phone line was installed, and the necessary teleprocessing software was tested and found to work beautifully. Then the problem was discovered — scheduling. The teleprocessing software used to read the library's terminals was incompatible with another teleprocessing system which was always scheduled to run until 8:00 p.m. The library's teleprocessing program couldn't be run until afterward. The process of reading the terminals' memory is normally automatic, but human assistance is required whenever anything goes wrong. The library had assumed that the terminals would be read near the close of each day, around 4:30-5:00 p.m., when someone would be available to monitor the operation. Instead, the staff now goes home knowing that one of them will be called to return to the library if a problem develops. On the average, the reading process is reliable, failing only about once a month. But staff members have been called out of bed at midnight to return to the library to push buttons on a malfunctioning terminal while diagnostic tests were performed from the computer center.

Agreement for Personnel Services

Libraries dependent upon centralized analysts and programmers must negotiate these services as well as machine services. The most important point to be decided is the basic arrangement by which development

personnel are assigned to the project. There are two models. In the first model, called the finished product model, the computer center promises to deliver a specified product on a certain date. The number and kind of personnel assigned can vary from week to week or day to day. The library is not concerned about the number of personnel assigned, and in fact, may not even know what the staffing level is. The computer center may assign only one or possibly two people to communicate with the library, while an unknown number of people work on the project "behind the scenes." In the other model, called the time and materials model, no fixed dates are promised. Rather, the library is promised a level of effort, usually expressed in terms of the number of FTE staff to be assigned for the duration of the project. For example, a project may be assigned two programmers who work full-time until the project is finished. No deadline is set, but a project completion date is usually estimated to give computer center management an idea of when their personnel will be available for other projects, and to give the library an idea of when they will need to be ready for the new system.

The first model sounds more advantageous to the library, but in actuality it is not. The first model is really the second model in disguise. When the computer center works according to the first model, their personnel meet with the library several times to become acquainted with the proposed project. They return to their offices and make an estimate of the magnitude of the project, usually expressed in man-months. Next, they decide what personnel resources would be available to work on the project and for how long, and finally they calculate a completion date. Unfortunately, calculated completion dates, which are nothing more than estimates, are promised as firmly committed dates. Ninety-nine percent of the time, the dates will slip. The commitment is not firm in the sense that if the development schedule slips, the computer center will add more people to the project to get it back on schedule. Computer centers generally do not have enough personnel to move around in this manner. The University of Illinois example mentioned at the beginning is probably an unusually bad case, but any library which makes plans for personnel, equipment or building modifications based on a "firm" date may incur extra expense or inconvenience when the date slips.

It is far more practical to get a commitment for a fixed number of personnel for the extent of the project. Completion dates are regarded as they should be — as estimates and nothing more. No one is deluded and, for reasons described below, the library is in a much better position to make sensible decisions as the work proceeds.

Of all the areas listed in Table 1, one of the most important to the computer center is an agreement on change control. Change control refers to a set of rules (admonitions, really) which should guide the library

when it seeks to make changes to the system during the development phase. The first rule indicates that the functions to be performed by the computer be specified correctly in the first place, so that no changes will be needed. The second rule is to get it 99 percent correct the first time so changes will be minor. The third rule is to do without overlooked functions until a later phase, when a whole set of improvements can be made at once. The fourth rule states that since the first three rules won't be followed, make change requests known as soon as the need is discovered and be prepared to accept compromise. Some changes can be made easily with a minimum of delay. More commonly, changes cause substantial delay and increase development costs. The library should be willing to agree to a clause on change control which adopts a philosophy that it is better to get a limited system running and gain practical experience than to request changes which have minor benefits. Projects always encounter a genuine need for some changes during development. The purpose of a change control clause in the agreement is more for psychological impact than procedural structure. The library needs to be warned in advance to be thorough at the outset and to restrict its demands during development to the very essential changes.

Documentation is of several types, two of which can be at issue. Types of documentation include functional specifications, system specifications, program documentation, production documentation and user manuals. The computer center is obviously responsible for system specifications, program documentation and production documentation. But functional specifications and user manuals can be the responsibility of either the library or the computer center, and this should be determined by agreement. Since a project cannot begin without functional specifications, agreement on responsibility is negotiated early. Nevertheless, at least one library forgot to discuss the user manual. The library assumed the computer center would write it; the computer center assumed the opposite. A lot of finger-pointing and unnecessary irritation resulted when the misunderstanding was discovered near the end of development. Implementation of the system was delayed six weeks while the library wrote a user manual.

Acceptance testing by libraries is frequently done superficially. The programmer tells the library a program is working. If it is a batch program, the programmer brings the library some demonstration output. The library looks at the program's reports to verify the presence of required data fields and check for bad data. To test an on-line program, a librarian will sit down at a terminal and enter data — both valid and invalid. If a program identifies invalid data while taking action with valid data, it is "accepted." None of these cursory tests constitutes a thorough

acceptance test, because the data samples are too small. One can only say that a program works for the combinations of data with which it has been supplied. Acceptance testing should be performed by a team of experts within the computer center who are not part of the development team. A separate team would have necessary expertise but would avoid the conflict-of-interest situation which occurs when a programmer judges the acceptability of his or her own work. Most computer centers do not use such teams. The programmer's test, occasionally supplemented by library tests, are the usual acceptance tests. Libraries will probably never be able to negotiate thorough acceptance testing. But the library and computer center can and should still make an agreement on a sign-off procedure for the cursory tests. Thereafter, the library should realize that the first six months of operation of a new system will be the real acceptance test.

Program maintenance and program extensions are two areas in which the computer center has much more experience than the library. The computer center may have a policy or procedures manual that documents procedures to be followed. The manual, with modifications if necessary, can serve as the agreement for these areas. Program maintenance here refers to the correction of programming errors discovered after the system becomes operational. Corrections need a high priority. The computer center should agree to fast correction of errors and should be prepared to assign someone immediately to the problem, even if it requires taking a person temporarily from a development project.

Program extensions are design changes made to a system after it is operational. If a system was designed to be fairly complete at the beginning, changes will not be major — but this is usually not the case. The library should have a commitment from the computer center to incorporate extensions over a period of time. Because it is more effective from a programming point of view to make several changes at one time rather than one change at a time, the most pragmatic approach is for the library to accumulate ideas for improvements and to assign a priority to each. An agreement between library and computer center on program extensions entails an obligation on both sides. The library is obligated to batch requests instead of asking for one change at a time, one after the other. The computer center is obligated, after a period of time, to program the top several features which will most improve the system. One set of changes might be programmed, for instance, at the end of six months of operation, and another set at the end of twelve months.

A variation on the time and materials model for assigning analysts and programmers is possible, and it has ramifications for all of the other areas of personnel negotiation. Rather than assigning a fixed number of personnel for the duration of a project, the computer center can be asked

to assign a fixed number of personnel for an indefinite length of time. The library, of course, would have to justify the request on the basis of some coherent long-range plans for automation. But if the case can be made for a long-term commitment, the need to negotiate a myriad of other details diminishes considerably.

The commitment will be for a fixed number of people with certain skills. The computer center does not have to promise specific individuals, but the arrangement works best when individuals can be found within the computer center organization who are interested in library data processing and will stay on the assignment. The benefits of such an agreement to the library are considerable. First, the library can explain library operations to the data processors in more detail, because the knowledge will carry over to later projects. Librarians who have seen a succession of computer center personnel, quickly tire of repeatedly explaining basic operations. Secondly, there is no need to negotiate formally change control, documentation, acceptance testing, program maintenance or program extensions. The computer center people are available to help the library in areas where help is needed most. The library can set priorities with full knowledge that more time spent on documentation will mean less time available for acceptance testing, or that time spent on extending an old system means time away from building a new one. Thirdly, computer center personnel know they will have to follow up on their work. If they do a poor job of testing, they will soon be required to do program maintenance.

Limitations of Agreements

Without a contract in the legal sense, there is no legal recourse to the solution of problems, and *a fortiori*, there are no penalty clauses. Agreements are like treaties; they can and will be broken. When they are, the problems must be solved inside the organization.

There are two possible places to appeal within an organization. The first appeal is to the computer center and, within the center, to the direct source of the problem, i.e., the computer operator, the programmer, or the systems analyst. Failure to solve the problem at the operational level requires the library to move its appeal up the hierarchy to computer center management. In most cases the problem can be resolved somewhere between the operational level and the computer center director. If the dispute reaches the computer center director, the library director will undoubtedly be involved.

When the problem is not resolved within the computer center organization, the second appeal is to higher administration. One may have to go up the organizational ladder until an administrator is reached who has jurisdiction over both the library and the computer center. In some

organizations, the administrator may be several ranks removed from the library and computer center. In universities, it is not uncommon for the computer center director to report directly or indirectly to a vice-chancellor for administrative affairs, while the library director reports directly or indirectly to a vice-chancellor for academic affairs. The first common administrator is, then, the chancellor, the chief presiding officer of the university.

The library may encounter a number of problems in its appeal to higher administration. In the first place, higher management may, in general, regard the computer center more favorably than the library. The computer center may be looked upon as a unit devoted to modernization and an aid to institutional cost reduction, while the library is seen as a traditional and ever-increasing drain on funds. If this is the case, the library appeals from a position of weakness. Secondly, the more the dispute is embedded in computer technology, the more predisposed higher management is toward the computer center. In theory, if not in fact, computer center personnel are the experts on technological matters and librarians are not. Thirdly, disputed issues are likely to be too detailed to generate serious attention from higher management. Can the library effectively protest to a chancellor that response time is seven seconds when the computer center promised it would be three seconds? The head librarian is apt to get a pat on the head and be told to worry about book budgets or building plans, but not petty details. In short, unless the library is in high standing with the upper administration, it has a better chance for enforcement at the level of the computer center, particularly when the agreement clearly identifies the computer center's obligations.

There are other limitations to agreements that are potentially as serious as breach of agreement. Problems can arise that are outside the scope of agreements negotiated by the most diligent libraries. It is erroneous to assume that all possible problems can be anticipated and incorporated into an agreement. Some issues will be overlooked, and hence will not be agreed upon in advance. Other problems can be anticipated, but for one reason or another, cannot be negotiated. In many institutions, several areas of machine services cannot be negotiated, because computer centers often provide a set of basic services for all customers and adopt a take-it-or-leave-it attitude. As long as these computer centers have enough business, they refuse to negotiate special services.

Higher administration may itself be the source of problems. Higher administration may cut the computer center budget, reducing the center's capacity to serve the library with machine resources or personnel resources or both. Higher administration may, alternatively, cut the library's budget,

reducing the library's ability to pay for planned automation. At one library, a serials list which had been started three years earlier was terminated for this very reason. The higher administration may decide on a top-priority computerization project which preempts a library project, even though the commitment to the library by the computer center had been a firm one.

The library cannot negotiate a clause stating that no human error will occur. By far the most common source of errors and problems in well-tested computer systems are human mistakes committed in the course of routine production operations. Jobs are forgotten and not run. Jobs are run late. Jobs are run on time, but the output is delivered late or is delivered to the wrong location. Preprinted forms get out of alignment on the printer, and the operator doesn't notice. Data can be mishandled by personnel. On one occasion, a large amount of data was keypunched and given to a courier who temporarily left the boxes of punched cards on a computer center hallway floor while he went into an adjoining room to speak to some colleagues. At this computer facility, scrap cards, being too voluminous for wastebaskets, were left in hallways for custodians to take away. The inevitable happened. During the few minutes that the courier had left his cards, the custodian came by. The cards were found in a large outdoor trash bin, wet and damaged.

The wrong files can be set up for a program, which then runs perfectly except for the fact that it is using the wrong data. One year, at the end of May, a computer center scheduled a normal program run in order to post the library's May book purchases and receipts to its year-to-date master file. The April year-to-date file should have been submitted to the program as the latest master file. Somehow, a computer operator selected the March year-to-date file. When the May fund reports arrived at the library, bibliographers were surprised, but pleased, to find they had more money than they expected. Coincidentally, free balances were about the same as the previous month's. Being near the end of the fiscal year, and as good bibliographers should, they all began ordering heavily to encumber the remaining money.

Worst of all, data can be completely lost. A typical scenario goes as follows. Two jobs are to be run back to back. The first job copies a master file from disk to tape for later processing, and the second job erases the disk copy of the master file to free the disk space for other use. The operator must not run the second job if the first job does not complete successfully. The operator fails to catch an error message from the first job; job two is released, and a portion or all of the data are permanently lost. There are more complicated variations of the story, but the results are the same — human error can result in the loss of data.

Solution and Conclusion

The library can take a number of steps to minimize the difficulties that may arise when it uses the machine and personnel resources of its organization's computer center. First, the library should get as much prior agreement as possible before starting on a project. If the library is already well into one or more projects, it is still possible to negotiate points not covered or renegotiate points by means of amendments. The essential purpose of agreements is not to assign blame when disputes occur, but to avoid misunderstanding in the first place by providing the computer center with a clear and complete itemization of services needed and expected.

If personnel services are supplied by the computer center, a commitment of a fixed number of people for an indefinite period of time is superior to any other arrangement. This arrangement comes close to actually having the personnel on the library staff, and it allows more continuity, improved understanding of the library, stronger motivation to do quality work, better communication and more flexibility to handle urgent unforeseen tasks.

In the area of daily operations, the library should negotiate the best schedules possible, insist on adequate backup and recovery procedures, and have library staff members examine output promptly and carefully in order to identify problems early. One should be prepared to communicate production problems quickly with adequate informational details — the absence of which makes it extremely difficult for computer center staff to diagnose a problem. If programs are written by the computer center, a clause should be negotiated which says that once data are in machine-readable form, they are the responsibility of the computer center, so that lost data which cannot be recovered by programmed recovery routines will have to be reconverted by the computer center without cost to the library. Privately, the library should plan for the worst, be ready to complain loudly and, if so inclined, seek divine assistance.

Formation of a users' group can be another effective step. Chances are very good that other units using the computer center will have similar requirements for good service and will be experiencing similar problems. The formation of a computer center users' group to discuss common problems and to make recommendations to the computer center can result in improvements that are not possible to negotiate individually. The users' group should have official recognition from the upper administration of the organization, in the same way that libraries have official advisory committees.

The last step is to try to solidify the library's standing with upper management. This might be accomplished by playing golf with them,

joining the same church or civic club — whatever works. Since appeals are not made in a court of law, there is no rule which prohibits the "adjudicator" from being partial to the library.

A negotiated agreement by itself will not eliminate all the problems associated with use of computer center services, even though it is clearly written and comprehensive. An agreement can't even guarantee the elimination of major problems, but it will reduce the number of problems. An agreement will improve relations with the computer center, smooth development and operation of computerized systems, and reduce staff time spent on problem solving. An agreement is not a panacea. It is, however, an important element in the library's successful use of its organization's computer center.

REFERENCES

1. Vdovin, George, et al. "Computer Processing of Serial Records," *Library Resources & Technical Services* 7:71-80, Winter 1963.
2. Bosseau, Don L. "The University of California at San Diego Serials System — Revisited," *Program* 4:9, Jan. 1970.
3. Grazzino, Tony (Acting Director, Administrative Data Processing, University of Illinois). Memorandum to Lucien W. White, University Librarian, July 3, 1973.