ON THE ROLE AND FUNCTION OF A USER SUPPORT GROUP
FOREWORD

This Note is a revised version of the three-part exercise "Towards a Definition of User Support" which appeared in October and November 1975. Aside from re-organisation and the excision of some redundant material, the major change is an increased emphasis upon the responsibilities of the User Support Group to represent the users' point of view within the computer centre. I would like to thank all those who took the time to read and comment upon the earlier versions, especially CERN's User Support Group.
TABLE OF CONTENTS

Introduction .................................................. 1

The User Interface ........................................... 1

The Needs and Desires of the Users .......................... 3
  Resources
  Attributes of the system
  Psychological requirements
  Exchange of information
  Other Needs

A Functional Description of the Computer Centre .......... 10
  Hardware
  Software
  Certification
  Education and assistance
  Resource allocation
  Monitoring
  Accounting

A Computer Centre Model .................................... 13

The Role of the User Support Group .......................... 15
  Hardware
  Software
  Certification
  Education and assistance
  Resource allocation
  Monitoring
  Accounting
  The user interface
  Summary

User Support and the Rest of the Centre .................... 25
  Critic
  Governor
  Interpreter
  Relationship with other groups

Appendices:

I: Loose Ends
   User-produced code
   A distributed centre

II: An outline of the duties of a User Support Group

III: Improving the interaction between User Support
    and other computer centre groups
On The Role and Function of a User Support Group

Introduction

This document is based upon the philosophy that a computer centre is a service organisation; that it has no purpose of its own other than to provide the best facility possible within the financial, administrative, and political constraints imposed by the parent organisation; that its value is determined by its value to its users, and hence is measured by the quality and efficacy of the user interface and not by traditional measures such as computing power, or "availability", or CPU utilisation, or average response time, or jobs per day.

From this basic premise it follows that every member of the computer centre staff is involved with "user support"; why then a separate User Support Group? Because that which is everyone's responsibility becomes no-one's; and because those involved with the daily tasks of running the centre find themselves continually confronted with more immediate, if not necessarily more important, problems. The principal responsibility of the User Support Group is to ensure that concern for the user is not buried in the welter of daily detail.

The intent of this note is to provide not only a description of the duties and responsibilities of a User Support Group and its relationship with other segments of the computer centre, but also to furnish some background and motivation for that description via a discussion of the breadth and nature of the user interface, the needs and desires of the users, how they relate to the functional organisation of the computer centre, and the means by which the computer centre can attempt to satisfy them.

Those who consider such exercises as merely academic and of no practical importance will find the "practical" content of this note distilled into the final two appendices.

The User Interface

An interface is an area of contact between two dissimilar entities. Humans and computing systems are as dissimilar as any two entities imaginable. Both are complicated entities; as a result, the interface between them is very diverse and complex. It includes all the ways in which users can interact with elements of the computing system*, whether those elements be people, machines, programs, output, documentation, procedures, or policy.

* In this note the term "system" refers to the whole of the computing complex; when only the basic software is intended, the term "operating system" is used.
An exhaustive list of all the elements is not practical; a partial listing, however, will illustrate the number and variety of the specific user/system interfaces which exist:

FORTRAN II (IV,G,H); WATFOR,WATFIVE;RUN,FTN; MORTAN
COBOL;ALGOL;PL/I;BASIC
JCL
LGO;LGOB;LIBEDIT;EDITLIB
Manufacturers' documentation
Local notes purveying corrections to manufacturers' documentation
Local program documentation
Local programs
Newsletter
Bulletin
I/O stations (central and remote)
User operated devices (card readers, printers, keypunches, etc; terminals, graphic or not, with telephone or not (including the operators' consoles))
(Informative) displays
The telephone system
Human interfaces (consultants, advisory committee members, I/O personnel, personal contact with individual programmers and managers)
The charging algorithm
Reports (if any) concerning resource usage, plans, etc.
Program Library (including compilers, assemblers, etc.)
Operational procedures
Operating systems
Printer plots; microfilm; plots on plain (lined) paper
Mikrofiche
Magnetic tape
Octal (hexadecimal) dumps

This great abundance of points of contact between the system and the users engenders in the latter a correspondingly great abundance of desires and expectations concerning the system. The totality of those desires and expectations imposes a complex and self-contradictory set of pressures upon the computer centre.

A successful centre is one in which the needs and desires of the users are nicely matched by the facilities and capabilities of the system. There are two methods by which this matching process can be pursued. The first is the traditional computer centre approach: to shape the people to the (intransigent) requirements of the system. The second is beginning to emerge in a few centres, and is the one advocated here: to shape the system to the (shifting) needs and expectations of the user community.

In such a centre, the user interface is always changing, and change is one of the things which users do not like. The User Support Group are the custodians of the user interface, and are charged with the responsibility of regulating the rate and direction of its changes; of seeing that it does not shift disruptively, and that all movements are designed to bring the system into closer conformance with the needs of the users. A prerequisite for success is an understanding of those needs.
The Needs and Desires of the Users

In an attempt to simplify the consideration of the users' needs and desires they have been grouped into a few broad categories: resources to draw upon; attributes of the system; psychological requirements; exchange of information; and a few others which fail to fit into any of these. Some duplication will be apparent: the same or similar items will be considered under more than one category. This is not accidental, but is intended to encourage one to examine those items from more than one point of view.

The resources of the system are the things which do the work the user wants done, or which enable him/her to use the things which do the work.

The attributes of the system are the qualities it must possess if (s)he is to work confidently and effectively.

The psychological requirements (and the other needs) are generally concerned with equipping the system to deal with the imprecision and variations in temperament and ability which characterise human endeavour.

Resources

The basic computer centre resources are

- raw computing power
- storage media
- ancillary hardware
- library of software
- reservoir of past experience
- access to the other resources
- work space

They are the fundamental tools and machinery furnished by the computer centre and as such are relatively straightforward. The largest problem they offer (except in cases where they are inadequate to the job) is the competition for access which arises from their scarcity.

Raw computing power: A calculating engine capable of doing a reasonable (self-contained) portion of the user's job in a reasonable time. (For most jobs, "reasonable" means less than an hour of dedicated time.)

Storage media: Enough on-line dynamic storage to permit reasonable structuring of the user's work. Working storage for programs-in-progress, including storage for test data. Storage for use during data accumulation phases. Storage for the retention of programs and data which will be used more than once. Archival storage.
Ancillary hardware: Input devices suitable for the quantity of data and its speed of acquisition. Output devices suited to the form (and, for pictorial output especially, quality) required, quantity delivered, and rate received. Input and output stations which support the style of operations desired by the user (remote input/output stations; user and operator-operated card readers, terminals, tty and graphic; user and operator-operated printers...). Job and system-status polling devices.

Library of software: Processors for (at least) the commonly used languages appropriate for the kind of work the user is doing. Cross-processors for the various small computers in use. Commonly used subroutines and utilities. Graphics packages. Permanent file subsystems. Application packages or subsystems for common processes. Facilities for building, maintaining, and using private libraries (source and binary)... 

Reservoir of past experience: (In addition to the software library): Accurate knowledge of the capabilities of the system, and how to invoke them. The use, when reasonable, of the programs and data of others. Instruction in the effective use of the system. Instruction in the inexpensive use of the system. A comprehensive set of debug tools. Expert assistance (of arbitrary extent) when needed.

Access to the other resources: (It is often necessary to control the access to some resource as tightly as the resource itself is controlled; guaranteeing an interactive user 10% of the CPU is worthless unless he also has access to a connected interactive port about 90% of the time.) A reasonable way to acquire routines from the software library. A port or station which is operable when the user wishes to compute. Procedures for the acquisition and retention of storage space (on-line and off). Reasonable schedules for output devices.

Work space: At each terminal, space for the user to spread out his material. (Temporary) storage space for visitors. Thinking space at all I/O stations. Quick-program-change areas at all I/O stations.

Attributes

The qualities the users want to see in a computing system are:
- accuracy
- dependability
- stability
- availability
- responsiveness
- speed
- continuity
- flexibility
- consistency
- predictability
- tolerance
- convenience
- simplicity
- informativeness
- helpfulness
They occur here in my estimate of the order of their overall importance, although different user communities (or even the same community at different times) are expected to have differing priorities. Whatever the relative priorities, it is doubtful that many users are satisfied by any computer centre in more than one or two of these areas.

**Accuracy:** Those operations which one does manage to get the system to perform must be done correctly: it is better not to do it at all then to do it wrong. (This includes the manual operations involved as well as the technological operations.) The system should never lie to the user; it especially should never say -- or imply by silence -- that all is well when in fact it is not.

**Dependability:** Schedules must be published, and published schedules observed. The system must complete whatever tasks it undertakes: all jobs input should eventually be run; all requests for information should be answered. System failures will obviously slow me down; they must not destroy me.

**Stability:** Breakdowns should be rare. Trivial changes should not cause catastrophic breakdowns. Overloads may cause halts: they should not cause crashes.

**Availability:** The system must be usable when the user expects it to be. It must be available a reasonable fraction of the time, at reasonable times, and for a reasonable duration of time.

**Responsiveness:** Discovered problems should be speedily fixed. Limitations should be quickly overcome, whether they be in kind or quantity of service desired. Questions should be answered (courteously). Requests for information should be honoured. Comments should be acknowledged and any resulting action should be advertised.

**Speed:** All tasks entrusted to the system should be completed as quickly as possible. (If the system is unable to complete a task it should immediately so inform the user.) (This heading thus includes CPU, channel, and device speeds; response time; turnaround time.)

**Continuity:** A job which ran yesterday should run today. Information which was available yesterday should be available today, in the same place, and in the same format. A capability which existed yesterday should exist today, accessible in the same manner.

**Flexibility:** The development of new applications should not be hampered by system constraints. Minor changes in requirements should not require major changes in design. Reasonable new variations should not cause unreasonable contortions. (It should, for instance, be trivial to change the output medium (paper, film, or CRT) or file residence medium at will.)

**Consistency:** Similar things should be done in similar ways. Control syntax should be constant across all job (step) input mechanisms. Different instances of identical conventional names should have identical interpretations. Don't give conflicting advice in different places. Eliminate gratuitous differences in different systems.
Predictability: The natural interpretation of a control statement should be the correct one. Avoid clever names: they obscure the purpose and confuse the uninitiated. Avoid mis-leading names. Obvious generalizations from familiar actions should give the expected result. Uniform (but predictable) mediocre performance is usually preferable to performance which is occasionally superb but frequently dismal.

Tolerance: Forgive forgiveable errors. Speak gently to interactive users; when in doubt, ask. Remember that the purpose of the system is to solve the user's problem, not to teach the user to become a computing expert. Accept obsolete syntax as long as possible.

Convenience: Facilities should be available at all reasonable hours (this includes such non-hardware facilities as consultants, receptionists, and libraries and librarians as well as the hardware). Minimize the number of reasons for which a user must apply in person to the computer centre. (E.g. make documentation and library programs automatically (and remotely) available; provide control card procedures for tape library transactions.) Provide the common languages. Simplify administrative procedures.

Simplicity: Simple syntax for (apparently (to the user)) simple tasks. Avoid circumlocution (don't require the user to say COPY (to some data sink) when he really wishes to SKIP). Avoid error-intensive syntax (RUN,,AX.) Provide reasonable defaults (which can be over-ridden in a straightforward fashion). Provide both global and local defaults. Minimize red-tape. Eliminate extraneous procedures.

Informativeness: Tell the user what's going on (in the large as well as the small). Avoid meaningless or mysterious comments. Put time and date on all messages, whatever their source. Let the user know what resources his job used. Let the user know which resources are (not) available. When response is slow, tell the user why. Tell the user what's coming, and what's going away (and when). Allow the user to inform himself.

Helpfulness: When a user asks for help, give it to him. Make error messages specific. Provide dump analysis (automatically as well as personally). Suggest alternatives. (Don't just say "RUN is going away", tell the user what to use in its stead.) Provide samples. When a user reports a problem don't simply explain it: help him to get his job done.

Psychological Requirements

This is the general area of system design, development, and implementation which is most neglected today. Much more thought and effort has gone into the efficient use of the computer than into the efficient use of the person using the computer.

Nevertheless, we can identify a few broad areas of concern:
- human engineering in interface design
- the human as master, the system as servant
- the elimination of unpleasant surprises
- the re-personalisation of the user

These are all based upon the (somewhat belated) emergence of the notion that the users are more important than the system, and that if the two do not get along comfortably, it is the system which should change.

Human engineering in interface design: System designers should recognise that people are not automata and do not always return the same response to a given stimulus. Different users have different abilities, different tolerances, different expectations; they proceed from different assumptions, arrive at different conclusions. They differ in all these respects not only from each other but also from system designers. Computing system interfaces must be designed with these vagaries in mind. A few reminders are listed below. (It is not thought to be a complete list.) Not all of these suggestions are relevant to all interfaces, but they all apply wherever they are relevant.

- An interface should be self-explanatory whenever possible: separate explanatory documents have a habit of becoming lost, out of date, and inaccurate.

- Interfaces, and the documents used to describe them, should be consistent, not only within the installation but also with the outside world. Thus, local variations of standard usage should be adopted reluctantly; similar verbs should have similar meanings wherever they occur; special characters should not vary from keyboard to keyboard; keystop graphics should agree with the character printed/punched/displayed or action performed, etc.

- Avoid jargon, especially in introductory publications.

- Use natural language insofar as possible. (Octal and hexadecimal numbers are part of no natural language.)

- Prolixity and excess concision are equal enemies of clarity; avoid both. All documentors should read "The Elements of Style" by Strunk and White.

- Let the punishment fit the crime. Severe errors may be punished severely, but don't let a single keystroke error destroy a day's work.

- Be polite, even under duress: it is the point of first contact which experiences the most friction. A soft answer turneth away wrath. Observe published schedules and timetables.

- Provide user control of all possible parameters: shape, quantity, format, and medium of output; level of competence as terminal operator; access to all possible facilities (even if dangerous (to the user himself; not if dangerous to the system or to other users) but not accidentally); timing and method of access;...
Avoid confusion/congestion in user spaces (I/O stations, terminal rooms, work areas, consultants' offices).

The human as master, the system as servant: This is an aspect of human engineering which is extremely important to some users. In general it means that choices should be provided wherever possible, and that whenever a choice is available the system should provide a method by which the user can direct it (always provided he accepts responsibility for his choice).

Some examples: the ability to direct work to a particular one of several "identical" modules, and to hold it up until that one is available; the decision at execution time whether to keep or reject a tape record with a parity error; ample warning when some system limit is about to be exceeded (with enough resources available to do something useful about it); the ability to replace a standard sub-routine or utility with one of the user's own devising.

Some of these processes are dangerous if misused, but they leave no doubt as to who is master.

Eliminate unpleasant surprises: Users have an investment in the present; most changes, even those which will eventually be to their benefit, cause them to lose some of that investment. As a result they tend to resist changes. Furthermore, in conformance with a well-known law of physics, the more abrupt the change, the more determined the resistance. Early warning allows time for psychological preparation and rational dialogue. It allows the user time to redirect work in progress. It is also common courtesy.

The re-personalisation of the user: A major complaint of many who must deal with large (computing) systems is the depersonalisation associated therewith. One way to minimize this feeling is to provide some personal services, especially when the user is in distress: someone to listen respectfully to his tale of woe (without dismissing him out of hand as an idiot: there is a difference between ignorance and unintelligence) and provide assistance; and a conscientiously applied program of investigating complaints, taking the action indicated, and reporting back to the plaintiff. Sensitive and intelligent personal attention can compensate for many technical inadequacies.

Exchange of information

Many computer centres do an adequate job of transmitting information from the centre to the users. It is equally important to see that some kinds of information get transmitted in the other direction. Of the kinds of information exchange discussed below, only the first is exclusively from-centre-to-user, the others are all either two-way exchanges or, in the last instance, one-way in the other direction.

- system documentation
- consultation services
- advance notice
- current affairs
- users' suggestions

All appropriate media should be used, both permanent and ephemeral, including personal visitation. Don't require too much formality: many good intentions have foundered in a sea of prescribed forms.
System documentation: All aspects of the system -- including its awkwardnesses, limitations, and constraints as well as its powers and capabilities -- should be adequately covered by user-oriented documentation (which recognizes that there are users of different levels of competence and experience). Documentation should exist, be accessible, and be accurate, complete, timely and readable. Documentation which is out of date should be destroyed. There should be, at all times, an accurate list of current documentation. There should exist a master index, covering all documentation. Cookbooks are godsend to inexpert users. A particularly important document for a site with many transients is the how-to-run-a-job-here guide. (It should include such administrative details as how to get a budget, an account number, a visitor's pass, working space, help, etc.)

Consultation services: Quick diagnosis and expert advice. (No guesswork: if a consultant doesn't know, (s)he should find out before answering.) Help with making programs run more quickly or more cheaply. Help with conversion problems (between successive local systems; between different local systems; between local and foreign systems). Liaison with systems programming and operations groups with respect to apparent problems. Uncovering pitfalls, limitations, awkwardnesses.

The above are all part of, or stem from, a service providing expert advice to the users. The complement to that service is the provision of expert advice to the centre: what the users are doing; why they are doing it. How they would react to proposed modifications and extensions. What they would most like to see fixed. When they plan to change their established patterns.

Advance notice: Any change in the external characteristics of the system must be communicated to the users in advance -- the bigger the change, the longer the advance. (And "bigger" in the sense of impact on the user; not difficulty or length of time of implementation: the removal of a compiler is technically trivial, but is a very big change for someone who depends upon it.)

Similarly, any marked change in the habits of the users, or in their requirements for any resource or service, must be communicated to the computer centre in advance -- the bigger the change the longer the advance. (Not least because satisfying the expanded requirements of one user may seriously inconvenience others.)

Current affairs: This includes such things as current levels of performance: what is turnaround like today for long jobs? How long will my permanent files last? How quickly can I stage a tape? Why is response slower now than it was 10 minutes ago, and is it going to get better or worse?

It also includes keeping the user informed of the current state and configuration of the system, and what parts (s)he is using: which mainframe and operating system, which tape drives, disks, printers were used; which significant elements are missing from the system; current system loading.

And the centre needs to know what sort of service the user is requesting, and what sort (s)he is actually receiving; what system elements are being exercised; which programs are used and which are shunned; how today's load differs from yesterday's.
User's suggestions: A great breadth and depth of experience lurks within the user community; one way to tap it is to solicit suggestions. Provide a working channel for the submission and processing of suggestions. If a suggestion is declined, tell the originator why.

Other needs:

- (Occasional) preferential treatment
- Protection from the mistakes of others
- Painless introduction to the system

(Occasional) preferential treatment: (Most) users are satisfied if they are convinced they are receiving their just share of the resources available. From time to time, however, circumstances arise in which a user's just share is not sufficient to accomplish the task at hand in the time available, but where a lion's share would allow successful completion. The system should be flexible enough to satisfy occasional lions.

Protection from the mistakes of others: My job and my files should be immune to accidental injury at the hands of other people or programs (including the system itself) unless I have explicitly delivered control into those other hands. (I specifically exclude consideration of malicious intent.) My job should not suffer because someone else exceeds a system limit. Whether or not the system should protect me from my own folly is moot; it should certainly protect me from yours.

Painless introduction to the system: There should be a nice, straightforward scheme for learning to use the system. If local folklore agrees that two out of three new users do THAT stupid thing, then steps should be taken to see that THAT stupid thing is difficult for new users to do. Cookbooks; signs; maps; assistance.

A Functional Description of the Computer Centre

As stated in the Introduction, the goal of the computer centre is to satisfy the requirements of its users as effectively and conveniently as possible. The basic tools with which the centre can attack the problem are hardware, software, documentation, allocation algorithms, personal dialogue, and administrative (political) clout. One might expect the functional description of a computer centre to parallel this inventory, but a somewhat different categorisation has been chosen, one which corresponds more closely to current practice:

- hardware
- software
- certification
- education and assistance
- resource allocation
- monitoring
- accounting
Hardware

The basic responsibility of the computer centre is the provision of enough hardware to get the work done in a reasonably expeditious and effective manner. How much is enough depends upon the variety and distribution of the various pieces as well as upon the power of each. (However good the rest of the system is, it cannot provide better service than the hardware is capable of supporting.)

Provision includes the maintenance of a state of readiness for use as well as acquisition. Readiness includes reliability: unreliable hardware is worse than none at all.

Hardware includes such ancillary equipment as terminals, microfilm and microfiche viewers, storage cabinets for various long-term and short-term (distribution or print-out, say) purposes, as well as the major equipment items. (In those cases where the acquisition is done by other groups the computer centre retains the responsibility to define accurately the specifications which should be met.)

Few user problems have pure hardware solutions.

Users can learn to operate a wide range of equipment, and often enjoy doing fairly menial tasks for themselves; however, users abuse equipment, sometimes deliberately (especially if it isn't working properly).

Software

Hardware alone is useless: software is required to transform it from a dead weight into a tool. The universe of software can conveniently be partitioned into five sectors; in order of decreasing intimacy with the hardware they are firmware, basic system and product set, utilities, application subsystems, (small) application programs. (Not coincidentally, that order also reflects decreasing involvement by the computer centre.)

Software is the source, simultaneously, of the great flexibility of modern computing systems and of the great frustrations so often experienced in trying to use them. It is that part of the total system with which the user is most intimately involved: a tool of almost infinite power and flexibility. Software limitations are accepted less gracefully by the users than hardware limitations.

The purpose of software is to make the full power of the hardware conveniently usable for the solution of the users' problems. Hardware and software may be substituted for one another to some extent.

Hardware engineering is reasonably well understood, and the importance of human factors is accepted (occasionally having some effect on design); "software engineering" is not generally accepted even as terminology, much less as a discipline, and the importance of human factors is both tentative and largely theoretical at present.

The responsibility of the computer centre with respect to software is to provide that portion which might reasonably be considered to be beyond the capability of the user to produce, or which is of sufficiently general utility to warrant centralised access. ("Provision" should be interpreted as for hardware.)
Certification

Neither the hardware nor the software is particularly useful
if it does not operate, correctly and accurately, in accordance with the
specifications. It is the responsibility of the computer centre to determine
the meaning of "proper operation", and then to verify that all pieces of
hardware and software for which the centre is responsible do, in fact, operate
properly.

Education and assistance

Despite our belief that one should not have to become an expert
informatician in order to use a computer effectively, it remains the case that
a modern computing system is an extremely complex entity. Its full power and
flexibility are not, in general, available to the naive, inexpert, uneducated
user. It is the responsibility of the computer centre to promote the effective
use of the system. A prerequisite for effective use is an informed and
knowledgeable user community. The creation of such a community demands the
existence of a variety of educational aids, most notably descriptive documentation
(at several different levels of detail) and individual instruction (usually
designed to overcome some specific difficulty).

One tends to think of documentation as rather formal descriptions of
programs and (if one is lucky) how to use them and what the output looks like.
Such things are included in the term, but it also embraces directories,
catalogues, indexes, signs, notices, posters, maps, diagrams, what to do, how to
do it, where to go, whom to see, cookbooks, examples, samples, books,
listings, newsletters, bulletins, writeups, pocket guides.

Individual instruction occurs most commonly as a by-product of the
consultancy. It should be always remembered that the good consultant learns
as much about the user as the user learns about the system ... and translates
that knowledge into recommendations for system development.

It also will happen from time to time that more general formal
instruction becomes necessary (when changing computers, for instance). The computer
centre is responsible for ensuring that enough courses, pitched at suitable
levels of competence, are provided.

Resource allocation

Although brief periods of prosperity are experienced from time
to time when the computer centre has more of something than the users want,
the normal state of affairs is one in which demand exceeds supply. To prevent
chaos, policies for the controlled allocation of (scarce) resources must be
defined and implemented. The definition of these policies is a political task
rather than a technical one, and is rarely left solely in the hands of the computer
centre. The centre is expected, however, to provide technical advice concerning
the capacities of various components, the current usage patterns, and the
feasibility of various alternatives. Implementation, of course, is the
responsibility of the computer centre. It must always be kept in mind that
the totality of the allocation and scheduling algorithms strongly influences the structure of the work entrusted to the system. All algorithms should be easily alterable to reflect the changing priorities and political relationships of the real world.

It is most often in this area, in fact, that political clout (power) is necessary: without it good ideas (and algorithms) do not always get implemented. It is necessary to adjudicate disputes in those situations where persuasion fails, especially when more than one group is involved. It is always the political realities which govern the allocation of truly scarce resources.

Monitoring

This is a second-order function in that it is undertaken not for its own sake, but in support of other functions. It should not, however, be allowed to fall on that account into neglect. The computer centre management, for effective planning for the future and for meaningful evaluation of the present, need regular, current, and accurate information on the state of the centre and the utilisation of its various components.

Accounting

Whether or not the users are paying real money for their computing they need to know the resources used by each job (accounting is different from billing). The various user groups need to know the amount of computing consumed by each project. The computer centre needs to know the activity of each user group.

This function is obviously rather closely related to the previous one (monitoring), but they are by no means equivalent.

A Computer Centre Model

The discussions to follow on the responsibilities of the User Support Group and its relationship with other computer centre groups can best be carried forward within the framework of a well-defined, if hypothetical, structure. The structure described below corresponds precisely with no computer centre of my acquaintance, though it shares elements with all of them. (It should also be emphasised that this computer centre model does not necessarily portray a model computer centre.)

The Stevens Hypothetical Computer Centre (SHCC) is comprised of six working groups plus management and staff. Four of the working groups are primarily concerned with products, two with services. The product-oriented groups are:

- Special Hardware Projects ("Hardware")
- Computer Operations ("Operations")
- Systems Programming ("Systems")
- General-Purpose Applications Programming ("Applied Systems")
The service-oriented groups are:

- System Utilisation Measurement ("Measurement")
- User Support

(Whatever substructure the groups may have is unimportant for this discussion.)

The general responsibilities of each of the groups are presented in the context of the functional description of the computer centre given above.

**Hardware**

Hardware responsibility is divided between the Hardware and Operations Groups, with the Hardware Group being responsible in general for local implementations and the Operations Group for acquisitions. The Hardware Group thus handles design and construction of locally developed hardware, and shares with Operations the responsibility for successful installation. Operations has primary responsibility for the installation, maintenance, and operation of all equipment, however procured.

**Software**

Some software is produced by each of the groups, but responsibility for the production, development, and maintenance of software is the primary task only of the Systems and Applied Systems Groups. The Systems Group is responsible for the design of local additions to the operating systems, and the installation and maintenance of all basic operating system and product set software, however procured. The Applied Systems Group has similar responsibility with respect to all applications subsystems considered general enough to be the responsibility of the computer centre. These two groups also share with User Support the development and maintenance of the software library, and the Systems Group may be called upon to implement some instrumentation code for the Measurement Group.

**Certification**

The actual certification of any element is the responsibility of the group which introduces that element into the system. User Support should be prepared to assist any other group in the preparation of a valid certification procedure.

**Resource allocation**

The establishment of policy is the responsibility of no single group, but rather of the computer centre management. The implementation of policy will fall to the appropriate group, usually Operations or Systems, depending upon the nature of the resource and the allocation algorithm adopted. The collection of data needed to administer the algorithm is the responsibility of the Measurement Group.
Education and assistance

User Support bears the bulk of the responsibility in this area, but should have the power to get assistance in the form of suitable (but perhaps preliminary) descriptive documentation from other groups as appropriate. In the event that suitable documentation is not provided, User Support should have the authority to deny an element incorporation into the system. ("Suitable" in this context means technically accurate and complete, but not necessarily conforming in all respects to internal standards for user documentation. Responsibility for this last remains with User Support.)

Monitoring

All groups need a reasonable amount of rather detailed information about current systems usage. The raw information is sometimes available, but sometimes requires additional instrumentation code to be added to the system. Whether this is accomplished by the Systems or Measurement group is a matter for arbitration or (friendly) negotiation. The "outer" layers of programming and the actual data collection and distribution are the responsibility of the Measurement Group.

Accounting

A specialized subset of Monitoring, with the same division of responsibility. Accounting is further complicated by the fact that the results go back to the users and, in those installations doing some form of charging, have financial implications; the human-engineering burden is therefore significant.

The User Interface

All groups are responsible for improving the user interface of the elements they introduce into the system. The User Support Group, as the Guardian of the User Interface, has the added responsibility of assessing the user impact of any proposed change and publishing a user impact statement before the event.

The Role of the User Support Group

This discussion is structured in the same manner as the presentation of the SHCC: first reviewing the responsibilities of the User Support Group with respect to each of the seven functional aspects of the computer centre, and then taking up the subject of the user interface.

Hardware

User Support participates only indirectly in the major hardware-oriented activities (procurement, development, maintenance, and operation), and then principally by providing the users' point of view in internal computer centre deliberations concerning hardware specification, configuration, and selection. The major User Support responsibility is knowledge: what the
hardware is, what it can do (speed, functions, capacity), what its limitations are, when it is available, how to get at it, etc. The User Support staff should also be aware of -- and communicate to the other portions of the computer centre -- those aspects of the hardware which the users find unsatisfactory for any reason. The User Support Group should assist in the preparation of meaningful user tests of new hardware, and should personally user-test all user-operated hardware. User Support should be alert for indications of subtle hardware faults, and should assist in their verification and isolation.

**Software**

Direct User Support responsibility in the software area is concentrated primarily in only one of the five types of software mentioned above (utilities). Although the software closer to the machine (firmware, basic operating system and product set) falls within the responsibility of the computer centre, User Support responsibilities in these areas are limited to the same kinds of responsibilities as those borne by User Support with respect to hardware; namely, the User Support Group is not required to install and maintain the basic software but to know what it is and how to use it, what it can do and how that differs from what it is supposed to do, what its limitations are and how to avoid or circumvent them. User Support, through the consultancy service, should be the principal agency of discovery of faults in installed software, and also the principal agency for the submission of suggestions for improvements.

The "outer" two categories of software (application subsystems and (small) application programs) fall somewhat further outside the domain of User Support than the inner two. User Support should have fairly complete knowledge of some of the application subsystems (those, such as graphics, which apply to the entire user community), but that requirement diminishes with generality, and only summary knowledge (the general kinds of work being attempted) of individual applications programs is required.

Before going ahead with the discussion of User Support responsibility with respect to utilities it might be well to define what is meant here by the word "utilities". A utility is a relatively small, self-contained program of (presumed) general usefulness. Many of the programs delivered with the basic system are utilities (file manipulation programs such as the COPSYS, for instance); most of the programs in the locally-developed program library are utilities. (Large utility-type programs are considered to be part of the operating system-cum-product-set if furnished by the manufacturer or to be application subsystems if produced locally or by a third party. Size is measured in terms of the maintenance burden: a program or system which requires a significant fraction of a dedicated person for continued satisfactory operation is "large".)

Direct responsibility (procurement, development, maintenance, replacement) for the utility programs is shared by the User Support staff and the Systems staff. The general rule is that those utilities which are supplied by the manufacturer, or which entail the modification of existing portions of the system, tend to fall into the domain of the Systems staff, while those which are add-ons tend to fall into the domain of User Support. (Under no circumstances should any attempt be made to establish rigid rules for this division of responsibility: it must remain amicably vague and be decided
pretty much on a case-by-case basis.) A pre-requisite for the exercise of this direct responsibility is a determination that a particular need exists. The User Support staff will be primary source of such determinations, whether the resulting utility is delegated to User Support or to Systems.

As with hardware, User Support should assist in the preparation of meaningful user tests for all software installed or supported by the computer centre, and should provide the voice of the user in internal computer centre deliberations with respect to software planning.

Certification

"Certification" is the verification of an (explicit or implicit) assertion concerning the system. When a new program is introduced into the library it carries an implicit assertion that it works correctly. Certification of a new (version of a) program or sub-system is the responsibility of the group which introduces it. User Support thus has direct responsibility for the certification of those programs provided by User Support. The User Support Group should also assist other groups in the development of meaningful certification procedures for their programs. (A case can be made for having the User Support Group undertake all software certification, but that can lead to severe conflict between User Support and the other software-producing groups. A more friendly approach is the joint development of a series of tests or criteria to be met.)

User Support, through the consultancy, hears many claims of suspect performance; these assertions, too, must be verified. One can frequently enlist the aid of the user affected but the responsibility for determining the truth of the situation rests with User Support. The consultants should take nothing on faith; the plaintiff has enough faith (in the correctness of his own actions) for all.

It is especially important that all documentation be accurate: that systems and programs act in conformity with their published descriptions; User Support is responsible for ensuring the accuracy of all documentation (whatever its source) relevant to hardware or software supported by the computer centre.

Education and assistance

(This is the generally recognized heart of the User Support function, the task, in fact, for which most User Support Groups were originally formed.) The kinds of help needed are many: analysis and help with the correction of programming errors whether committed by the users or the implementors; selection of proper subsystems, languages, utilities; selection of reasonable data structures and formats; the processing of data from incompatible systems; the conversion of programs between systems (domestic and foreign); the discovery of ways to circumvent limitations; the analysis of expensive programs; the recovery of lost or damaged data or output; the gentle introduction to a new environment; the timely performance of urgent work.
The best method of providing assistance usually is personal dialogue backed up with good documentation, and this is more true the less experience the user has. (Documents are better reminders than they are teachers.) Some user requests for assistance can best be satisfied by the production of additional software; this is not usually the responsibility of the user support group, but of the user himself (if it is of small general usefulness) or of the (systems, usually) programming staff. It is the responsibility of the User Support Group to know whether new software must be provided or whether something suitable already exists. Some user requests require administrative clout, especially those concerned with administrative or political limitations of the system, or with forcing work through it.

Technically competent personal dialogue is provided by a cadre of expert consultants, who have broad knowledge of the system in general and know its peculiarities and idiosyncrasies. It is extremely important that the consultants be dedicated in the sense that consulting be their principal activity and primary responsibility. It is not necessary, and in most cases is not desirable, that anyone be assigned full-time to the consultants' desk: consultants need time to follow up the problems and suspicions brought to them by the users, they need time to maintain their expertise, and they need a rest from the surprisingly heavy emotional burden borne by an active consultant. Thus an equally important responsibility of the User Support management is the assignment of the balance of the consultants' time to tasks which enhance their value as consultants and which do not interfere with either the consultancy itself or the necessary follow-up activities.

A major goal of the consultancy should be the creation of intelligent, educated users. (Intelligent users are harder to fool, and often exasperating, but they can contribute much to the improvement of the centre.) Therefore, it should specialise in the diagnosis, rather than the treatment, of unhealthy programs. It should encourage questions of the "How can I..." variety. It should be aware of which aspects of the system cause the most trouble and work to minimize the difficulty by changing the system, creating utilities, or educating the users.

A good consultant must possess a great amount of knowledge about the system: its capabilities and limitations (and how to get around the latter); all resource allocation algorithms (and ways of restructuring programs to fail within constraints and the circumstances under which the constraints may be removed); the interplay between charging and scheduling algorithms; the interplay between scheduling algorithms and job turnaround time; the relative reliability of various system components; service interruption schedules; future plans and how they impact various classes of users; current system loading; differences between the current system and its predecessors; differences between the current system and other systems (domestic and foreign) with which your users may be more familiar. Above all, (s)he knows where to get additional information or assistance when faced with a question or a problem (s)he is unable to handle.
It is frequently the case that the particular problem brought to the experts is the wrong problem, in the sense that it was encountered in the course of the wrong attack on the real problem: a good consultant becomes adept at detecting such false starts.

Consultants can also provide second-order assistance with respect to problems requiring additional software or administrative clout: they know to whom the user should turn and what procedures (if any) he should invoke. With respect to software, they are in a position to judge the general usefulness of the proposed package.

To be effective, the consultancy must be available where the users are, when their questions arise. This means through as much of the working day as possible, at the points of job submission and output distribution.

Although the consultancy is a most effective teaching device, it tends to be most used after a disaster of some sort. User Support also has the responsibility of helping the users to avoid these disasters in the first place. This is accomplished either through formal education and training (where practicable) or through various methods of documentation. Formal education is most useful when large numbers of people need to be taught the same thing; i.e. either when there is an influx of new users, or when a potentially disruptive change looms upon the horizon. During the ordinary course of events, the informal one-to-one arena of the consultants' office, supplemented by occasional educational articles in the newsletter, seems to work best.

As for the documentation, a wide range of media and styles should be used, depending upon the seriousness, urgency, and complexity of the message. Some devices which have been used, with more or less success, in various places are: dayfile messages to alert the users of obsolescent routines or those in the process of change; bulletins on the first or last page of all print-outs; broadcast messages to all connected terminals; closed-circuit TV or similar displays; a regular newsletter; an irregular newsletter; bulletin-board notices; user handbooks; pocket guides; on-line write-ups; general postal distributions;... Anything which works should be used; anything which does not should be abandoned (but not forgotten - the user community and the environment change from day to day, and what failed yesterday may well work tomorrow).

The kind of material appearing in any document is a function of the purpose of the document and the audience for which it is intended. Documents which describe existing programs and hardware need only be descriptive and may be rather technical; documents which describe future systems need more rationale and explanation; documents which describe how to use an element of the system or how to perform some specific act should be extremely specific and exact. Similar styles should be used for similar purposes throughout the organisation. Different documents addressing the same aspect of the system should be consistent.

Some descriptive documentation for each element of the system is usually provided by the manufacturer of that element. User Support should determine whether or not that documentation is suitable for use as it stands, or whether it needs modification or expansion. (It will also be, in general, User Support who accomplish that additional work.) User Support should insist
upon the availability of a certain level of documentation before any element is added to the system or before any user-detectable change is installed; this insistence will be effective, however, only if User Support (again) is willing to do most of the actual writing and publishing of that documentation.

All of the above has been concerned with educating the user about the computer centre; User Support should also undertake to educate the centre about the user; especially about what the user likes and dislikes about the current state of affairs, and what changes the user would like to see installed. A certain portion of this task is sometimes handled by an advisory committee at a relatively high level of management, but that level tends to be too high to admit of day-to-day experience with the system. The consultants get much more immediate - and much less politically-loaded - feedback about which aspects of the system are hard to understand, which have errors, which don't work as expected, which interfere with real work, etc.

It is not enough, however, to depend upon this by-product of the consultancy. The User Support Group should actively solicit the opinions and suggestions of the user community. Three avenues which can be pursued are a suggestion box (literal or figurative), general user meetings, and a visitation program. A certain amount of work is necessary to make any of these programs a success. The suggestion box and general meeting will fail if the user community is apathetic. The best way to produce apathy is to ignore user suggestions. Therefore, such programs should not be undertaken if the centre is not prepared to accept input from the users.

An alternative to a continuously open suggestion box is a broadcast call for input on some particular subject or aspect of the system: new features for the control language, for example, or permanent file retention algorithms. One should NOT expect an overwhelming response; users are in general not interested in computer science projects.

A better way to collect input from a spectrum of users is to call upon them personally. This has the advantage of reaching those who, for one reason or another, do not participate in the suggestion program or in user meetings; some of these quiet people make very worthwhile suggestions and comments. Such a visitation program should be handled with care -- the visitors must remember that they are ambassadors of the centre and may be considered as official spokesmen. They should also remember that it is more important to listen during such a visit than to speak.

The User Support Group is charged not only with collecting this input from the users, but with seeing that this information, perhaps recast into a number of specific recommendations, finds its way to those responsible for defining future projects.

Resource allocation

(This discussion refers to all resources utilised by the users: it thus includes storage cabinets, service schedules (for such personal services as the Consultants' Office as well as for hardware and software), library access, and the other ancillary resources as well as the CPU's and the various storage media.)
As has been noted before, resource allocation is primarily a political function and is usually entrusted to the computer centre only during periods of superfluity. During these periods, however, the computer centre should be acquiring data on usage patterns and trends so as to be properly prepared to make rational recommendations to the appropriate political body when the time of saturation arrives. The collection of raw data is a monitoring function and will be discussed in the next section; the determination of the reasons behind the facts (i.e. why people are using more (of whatever it is), the manner in which they access it, and what their plans are for the future) is largely the responsibility of the User Support staff. A closely related function is the determination of the effects of the various alternative proposals upon usage patterns, work habits, and job structuring. This arises most naturally out of a sound knowledge of the effects of present algorithms upon usage patterns, et cetera.

Thus, the User Support staff must know thoroughly the existing algorithms, the various dodges used to circumvent the limitations implicit therein, and how the users twist them to their own advantage.

The User Support Group also has the principal burden of the explanation of new algorithms to the users; it therefore behaves them to know why a particular algorithm was adopted and what it is expected to accomplish, as well as how it operates.

**Monitoring**

Many of the activities alluded to so far depend in some measure upon an accurate picture of what is actually happening with the users and the system. In most computer centres the responsibility for collecting the data for such a picture is quite diffuse: one group will determine usage of library routines, another the frequency of interactive interrupts, another the pattern of tape and/or permanent file usage, yet another will provide the input for the billing process. This diffuseness encourages a certain duplication of effort and guarantees fragmentation of the overall picture. Monitoring is not traditionally thought to be part of the User Support function, but (1) its exercise is certainly within the range of activities suitable for a consultant, and (2) its results should be well-known and understood within the User Support Group. Wherever the responsibility is placed, two things are essential: it should be centralised (all monitoring should be accomplished within one group), and User Support should receive all results.

**Accounting**

Accounting consists basically of two functions: collecting data about resource usage by individual jobs and by groups of users, and informing the users (and possibly some central financial authority) of the results. Most systems have at delivery some built-in mechanisms for collecting data on the individual job level and reporting it to the submittor, but every organisation has a different opinion about what should be accounted for and the manner in which it should be reported. Thus some work needs to be done on the gathering and formatting of the desired information; in short, the accounting task is a reasonably self-contained and well-defined subset of the monitoring task. It is therefore
desirable that responsibility for the accounting function go to whatever
group has responsibility for the monitoring function.

Whether or not this is the User Support group, User Support will have
an extremely important interaction with the accounting system: one of the
responsibilities of the consultants' office is the validation of requests for
refunds. One assumes that the refund policy will allow refunds only for
system faults; it is the consultants who will decide whether a failure is due
to system fault or user error.

The user interface

The primary responsibility for providing a good user interface for
an element of the system rests with the manufacturer of that element. Thus
User Support is responsible for ensuring that all of its products and
services (program library, utilities, write-ups, newsletter, consultancy, etc.)
have a good user interface. User Support should also serve as consultant to
the other computer centre groups with respect to the attitudes and expectations
of the users:

In those situations where either the manufacturer (coming from outside
the organization) or the element itself is so recalcitrant as to preclude the
development of a reasonable user interface, the User Support Group should under-
take to provide an intermediary layer which smooths out the rough spots.
(An excellent example of such a recalcitrant manufacturer/element combination
is JCL for the 360/370 computers; some installations let their users wander in
the morass essentially without assistance, while others have provided an
excellent intermediary layer of catalogue procedures.)

Another way in which the User Support Group can work towards good user
interfacing (with the assistance of sufficient political influence from upper
management) is the institution (and enforcement) of internal standards for
internal products. Thus, for instance, can uniform syntax be promoted and
adequate documentation; thus can awkward service hours be improved or unpleasant
user work spaces in the computer centre be made pleasant.

But before continuing a consideration of the general problem, let us
return to User Support's responsibility for their own products. The two
"products" with the greatest user impact are the consultancy and the
documentation.

The consultant's hat can be a very uncomfortable one. Its wearer is
continually facing unhappy, frustrated, angry people. They are people, by and
large, who have no interest in understanding the consultant's problems, but
who demand that the consultant show an intelligent interest in theirs. They
bear false witness, not only against the system, but also with respect to
their own actions. (This stems not from malice but from their general
unreliability as witnesses.) They have unreasonable expectations; and
occasionally become unpleasant when those expectations are denied. (On the
other hand, many of their "unreasonable" (from the point of view of the system)
expectations are really quite reasonable (from the point of view of the user);
the consultant must always be able to see things from both sides.)
They tend to attack the consultant for the deficiencies of the system. They are frequently ignorant and occasionally unintelligent. (The consultant must be able to discern the difference.)

As a consequence, the consultants must be selected with care. Mere technical competence will not suffice: it is not enough simply to answer the user's questions or to solve his (her) problem; the consultant should also contrive to teach the user something (to forestall future, related, problems); to learn something him(her)self about that user in particular and user problems in general; to communicate to the user a feeling of personal interest. This is extremely difficult, especially when facing the same trivial error the fifteenth time in the week after the cure was published in the newsletter. This is one reason why consultants are subject to early burnout unless they have intervals away from the office.

The consultants must try to remain emotionally detached from the system; any ego involvement whatsoever leads the consultant to construe anger against the system as anger against him(her)self. This is particularly true in time of stress.

The essential thing to remember with respect to documentation is that availability -- even delivery -- is not enough. The document has not served its purpose unless (a) it says what it was intended to say, and (b) its intended audience understands the message. The traditional virtues of accuracy, completeness, clarity, and timeliness address the first part of the problem; for the second, the centre must ensure that the users know the document exists and can obtain it when they need it. (One way to assist in the latter process is to keep as much of the documentation as possible available on-line.) This tends to mean to ensure that the users read the newsletter. There appears to be no foolproof method, but one must try. It seems to help if the format and style of the newsletter are tailored to the composite user personality. I have found that a chatty style works better than a formal one, and that a little bit of humour is welcome. Above all, be honest: things are not "going well", whatever the MTBI, if expected turnaround is three days.

Other problems result from various forms of over-kill: full coverage is necessary but if too much is sent none of it is read; when time is short, stridency is necessary, but if it happens too often it loses its effect; some technical vocabulary is necessary to achieve coherence and accuracy, but too much jargon restricts the effective audience too severely.

The first of these problems can be addressed by the general distribution of catalogues, indexes, and change lists instead of full documentation (as long as the full version is instantly available when needed); the second by insisting upon ample advance notice; the third is a function of knowing the audience for whom the message is intended and phrasing it accordingly.

You will notice that little has been said about who actually originates the various documents and messages whose reception the User Support staff is trying to encourage. User Support is rarely the true originator of the information, yet bears the responsibility for seeing that the information is distributed. This calls for a high level of cooperation between User Support
(the distributor) and the proper originator (who frequently is unwilling to spend the time to ensure a proper presentation of the material). Properly handled, however, this leads to willing contributors.

Now back to the larger problem, the care and improvement of the general user interface. The first step in dealing with this is to know what the interface is like, and what's wrong with it: know who is using the system and for what. Know the critical deadlines. Know what hardware is available. Know which components of the system are reliable and which are not, and know the nature of the unreliabilities. Know the limitations of the system, and which are firm (i.e. technical) and which are soft (i.e. political). Know to whom to apply for political immunity. Know the common sources of error and how to get around them. Know how tomorrow's system differs from today's. Know the promises which have been made concerning new system performance (and how well they are being kept). Know what just went down and how it is likely to effect the user. Know all allocation algorithms (including scheduling), and how they affect the design of the programs. Know the charging algorithm (and how to get around it). Know which portions of the system are overutilised or underutilised.

Knowing what's wrong with the system means somewhat more than a continuing analysis of broken programs with consequent reporting of the (user or system) error uncovered. The User Support Group should also serve as a detective agency with (at least) the following goals: detection of subtle faults (hardware and software) -- the ones which show up only under rare circumstances; early detection of trends (users tending to exercise some new facility more heavily; hardware failure rate climbing; etc.); detection of gaps in the system (perhaps with some suggestions for filling them); detection of ineffective usages.

The three principal methods for acquiring this knowledge are the visitation program, the consultancy, and the running of programs to test various facets of the system. A certain amount can be learned, especially with respect to long range trends, by the use of a suitable package of monitoring (hardware or) software. A regular periodic dayfile scan, for instance, can tell you a lot about trends in utility package usage. The User Support staff should also always be aware of whatever equipment (operating system) monitoring results have been collected by the Measurement Group.

Finally, the purpose of gathering all this information is to improve the lot of the users and to protect them from further abuses. It is only if they know what's wrong that the User Services Group can press for correction; it is only if they know which constraints are most frustrating that they can promote intelligent development.

There is a tendency among those who operate computer centres to believe that the computer (or the operating system) is an end in itself, and that the proper criteria for judging a centre are elegance, sophistication, high (CPU) efficiency, long MTBI, or any of a number of other worthy goals. In fact, the only proper criterion is utility (as seen by the users), and those others are useful only insofar as they promote utility. It is the responsibility of the User Support Group to see that this central fact remains central. They should examine all proposed "improvements" with a
jaundiced eye and ask "What's in it for the users?" Certain classes of system change are very disruptive from the user's point of view; user support should strive against such changes. (One particularly bad example is the irreversible change: a basic incompatibility which prevents a job being structured so as to run on both versions. Another is the removal of a well-loved utility. Another is the redefinition of some bit of control language.) Continuous paths can be provided, but they tend to be longer: fight for them anyway! Make sure new systems work before release.

Summary

The User Support Group has two general classes of responsibilities: the first is the provision of a number of products and services to the users, and the second is the performance of a number of services for the users. The services and products in the first category are an intelligent and sympathetic consultancy, suitable documentation, a comprehensive program library, and some of the utility programs, pretty much in that order of importance.

The services provided for the users include being the voice of the user in the deliberations (especially with respect to future planning) of the computer centre, safeguarding the interest of the users when change is being considered, insisting upon and assisting in the adequate testing of new elements.

Other responsibilities may be imposed (the monitoring and accounting functions, for example), but only if they do not interfere or conflict with the primary duties.

User Support and the Rest of the Centre

The User Support Staff plays three fundamental rôles vis-à-vis the rest of the computer centre: that of constructive critic, that of governor (in the relatively archaic mechanical engineering connotation of the word), and that of interpreter. An unfortunate adjunct of the first two rôles is that they tend to produce adversary situations, which in turn produce ritualistic antagonism, which sometimes hardens into the real thing. This antagonism must be avoided: it interferes with the cooperation necessary to ensure that the users get the best service the centre can provide.

Constructive critic

This rôle is in some sense the most "fun" because it involves discovering the faults in someone else's work. Usually the nature of the fault, and the fact that it is a fault, will be quite clear to everyone concerned, but there are often times when the evidence is ambiguous or the point a philosophical one. In the first case it should be the responsibility of User Support to remove the ambiguity, to provide a test or a situation which illuminates the true nature of the problem. The second is somewhat more difficult: there can be legitimate disagreements about relative priorities, or which of several alternatives is really in the best interests of the user, or whether
short- or long-term benefits are more important in a certain situation, or whether the cost (to the user) is worth the gain (to the user), etc. Some means of arbitration is necessary to resolve such impasses.

It is important, when acting as a critic, to avoid nastiness. Remember that those being criticised are usually those upon whom one must rely to correct the situation. It is neither necessary nor desirable for user support to attempt to fix the blame for an unfortunate situation whose cause lies outside of User Support: it is far more useful to get the situation corrected.

The procedures for reporting problems need not be excessively formal, but they need to be understood by the whole of the computer centre staff: who is authorized to submit problem reports, to whom they should be submitted, how they should be published (if at all), who should follow up, under what circumstances they may be closed. Similar procedures for the handling of suggestions are desirable.

**Governor**

This rôle is fairly simple in concept but difficult in practice. Its function is to keep the implementors from moving too fast; the difficulty arises in determining how fast is too fast. For the users any change comes too soon ... unless it liberates some stalled project. For the implementors, any delay after the thing is "done" is too long. One way to accomplish a slow-down is to provide a test which any system change must pass. (Keeping such a test current and realistic is a very difficult and time-consuming job.) Another is to require some number of prerequisites: n days' advance notice in a newsletter; prepublication of user oriented change documentation. However, just as change for its own sake is to be avoided, so is delay for its own sake. The establishment of a centre policy with respect to the frequency and magnitude of system changes may eliminate some of the friction between those who wish change and those who don't. (It should be remembered also that installations not writing their own software are somewhat at the mercy of the originator in this instance.)

**Interpreter**

The least controversial of the rôles user support must play: to explain the system to the users and the users to the rest of the computer centre. The accomplishment of the first part is the goal of the documentary activities of the user support staff, and in general the more help user support can provide the happier the implementors will be. The second part, again, can lead to conflict, for it encourages ritualistic antagonism, nevertheless, it is extremely important to let the voice of the user be heard in the councils of the implementors.

**Relationship with other Groups**

As in all interactions between people, some of the interactions between computer centre groups are relatively neutral, but many are (potentially, at least) emotionally loaded. Therefore, while much of the necessary dialogue can be carried on quite informally at all levels, some of it must be handled quite carefully and formally.
The kinds of interaction which might occur are:

- exchange of information about present projects and future plans
- requests for assistance, information, advice, opinions
- exchange of information about current performance and problems uncovered
- discussion aimed at influencing design
- discussion aimed at influencing priorities
- the partitioning of an area of shared responsibility
- the exercise of veto power

(It may be that the management of a real computer centre (as opposed to the SHCC) will reserve to itself the actual exercise of veto powers; that imposes only minor, and obvious, modifications to the discussion below.)

General comments

The User Support Group should be involved in all of these kinds of interaction (except possibly the last two) with all other computer centre groups. In order to keep the users informed about the state of the centre and its plans for the future the User Support Group must first get this information from the other groups. In the struggle with a particularly knotty problem a consultant needs to count on the assistance and knowledge of the experts residing in the other groups. When a consultant encounters evidence of hardware or software malfunction(s) he works with the appropriate other group to define and resolve the problem. User Support should be involved in the design of the user interface of all locally constructed (hardware or software) elements of the system. User Support should bring its knowledge of the state of the users, as gained through the visitation programs, the consultancy, and from suggestions, into computer centre discussions about the relative urgency of various projects.

Keeping this rather sweeping general statement in mind, we now look at the individual intergroup interactions.

User Support/Hardware

Most of the time, the User Support Group has only passing interest in the activities of the Special Hardware Projects group: most of the elements produced by that group are invisible to the user and affect her (him) only indirectly. Thus most of the intercourse between the two groups is the (regular) exchange of information about projects and plans. In those aspects of the work of the Hardware Group which affect the user directly (installation, testing, or maintenance schedules which affect user access to system resources, for instance, or the design of a bit of user-operated equipment), the User Support Group may be expected to take a greater interest. User Support should be at least informed (if not, indeed, consulted) in advance about every schedule change with visible effects on the user, and should concur in the design of the user interface of any piece of user-operated equipment.
User Support/Operations

(A more intimate relationship than with Hardware.) User Support should be consulted on the establishment of all service schedules and on all proposed non-emergency changes and interruptions. (And should be informed of the emergency measures as soon as they are known.) User Support should bring all suspected hardware malfunctions immediately to the attention of Computer Operations, and should assist in their verification. User Support should be involved in the design of the Operations-user interface: arrangement of user work areas, selection and location of user-operated equipment, procedures for access to resources (including the operation of user-operated equipment), distribution of output, format and methods of job and system status enquiries, etc.

User Support/Systems

(The most intimate of the relationships.) The Systems Programming Group's area of responsibility constitutes the bulk of the perceived interface between the users and the computer centre; the User Support Group consequently find the Systems Group to be the most interesting of the other groups. Most of the problems and limitations sensed by the users have software solutions; the result is a constant large backlog of possible projects. Users make (some of) their desires known via the visitation program, the consultants' office and the suggestion box. User Support needs an opportunity to pass that input on to the Systems Group. All changes to the software have implications for the users, some quite serious; User Support must be kept adequately informed of the nature of all planned changes, and the user impact thereof, sufficiently far in advance to provide the user with adequate warning and (if necessary) conversion aids. (This is one area where veto power might be invoked: if the user impact is not clearly specified, it doesn't go into the system.) The amicable distribution of the production of utility programs must be worked out. (And remember: the User Support Group must be involved in the design of even those utilities produced by Systems.) The principal burden of detecting software malfunctions lies with User Support; the more clearly User Support can state the nature of the problem, the sooner its resolution can be worked out. The preparation of an adequate test for an operating system is an unsolved problem; the User Support Group must assist the Systems Group in the development of an acceptable approximation to an adequate test procedure.

These two groups should be in daily contact so that User Support always knows that Systems is doing (and why) and so that the Systems programmers always know what the User Support staff would define as the major holes and awkwardnesses of the system.

User Support/Applied Systems

(Similar to User Support/Systems, but not quite so intense.) Although the products of the Applied Systems Group are general enough to be considered the responsibility of the computer centre, they tend to be intensively used by only a relatively few users. These users themselves, then, should be the principal folk with whom the Applied Systems Group interact. However, User Support has a responsibility to see that the usage, procedures, and even syntax
of these new elements do not conflict with the system as a whole. Therefore, Applied Systems must see that the User Support Group participates in the design of (at least the user interface of) all new elements.

User Support/Measurement

The User Support Group will be the Measurement Group's biggest customer, not necessarily in terms of new information requested, but in completeness of coverage: User Support should be copied on all information gathered by the Measurement staff. For the rest, except for the accounting function, the relationship between the User Support and Measurement Groups should be a fairly straightforward one: Measurement is a service organisation numbering User Support among its subscribers. Since the accounting function has a rather critical user interface, however, its design needs active participation from User Support.

Summary

The User Support Group is thus seen to be somewhat of a busy-body, continually poking into the affairs of the other groups and offering opinion and advice. Unless the whole of the computer centre staff understand and accept the real purpose of the computer centre, and understand and accept the rôle of the User Support Group as the Voice of the User, then that poking around is going to be resented as gratuitous meddling, and instead of cooperation the other groups will offer resistance. On the other hand, unless User Support handles all that poking about with great delicacy, the charge of meddling will be justified.

The three fundamental rôles which User Support must play with respect to the rest of the centre are rôles which tend to create conflict, and hence which carry the seeds of destructive internecine warfare. In order to avoid this unpleasant extreme it is desirable to establish rules of conduct: well-understood procedures for the announcement and verification of troubles, clearly defined responsibilities for various aspects of the system, courtesies to be observed etc. It is especially important to recognise that two groups with different responsibilities can arrive at irreconcilable positions and to establish rules of arbitration to cope with such situations.
APPENDIX I

Loose Ends

There are at least two topics which have so far been neglected. Earlier discussion would have complicated things a bit, and so they have been left for this Appendix.

User-Produced Code

In any large installation there are a number of extremely competent users who build utilities or applications packages which come into widespread use. The question of turning these over to the computer centre for integration into the system frequently arises; what responsibility, if any, does User Support have in such situations?

The decision with respect to such offerings should be made on the same basis as for any other software product under consideration: does the benefit justify the cost of installation and maintenance. User Support can help to assess the benefit and cost, but the decision should rest with the group which would assume responsibility (the Applied Systems Group for a package with significant technical content, and the Systems or User Support Group for a general utility). It may also be that some changes are necessary to bring the package into conformity with internal standards; the accepting group should feel free to require that these changes be accomplished before the package is accepted. (A similar statement with respect to documentation can also be made.)

A Distributed Centre

Assuming that the discussion so far is sufficiently general to cover both strongly centralised and highly distributed centres, does distribution present problems not explicitly touched upon? Of course the answer is "YES", because various logistical problems become more complicated, and because the User Support operation in general becomes more expensive and more difficult. The wide distribution of users means that more of the consultants' task must be done remotely (rather than in person); the existence of more than one centre of user activity suggests that there should be more than one consultants' office; methods of long-range access to the software library and documentation must be devised; the consultants must add additional areas of expertise to their knowledge of the system (terminals and remote communications). It is easier for users to feel disassociated from the centre and more difficult for them to affect its operation. The difference from a User Support point of view however, between a centralised computer and a distributed one remains primarily one of degree rather than kind.
APPENDIX II

An Outline of the Duties of a User Support Group

I. To keep the Users informed

It is the responsibility of the User Support Group to see that computer centre users are kept informed about the nature and state of the system; its features, facilities, capabilities, capacities, and limitations; known problems (and how to get around them); physical, technical and political constraints on system access and usage; how to use the system effectively and economically; plans for future expansion or modification. ("System", here and below, means total system -- hardware, software, people, and services -- not just the operating system software.)

Specific activities and services in this area include:

- regular publication of a computer centre newsletter
- Users' guides or handbooks
- maintenance, verification, and local modification of manufacturers' documentation
- preparation of user documentation for locally-developed software
- maintenance of the system status displays
- a computer science library
- general users' meetings
- formal classes, as necessary

II. To assist the users

It is the responsibility of the User Support Group to assist the users in the preparation of their work, in the understanding of erroneous or mysterious results, and in securing the (satisfactory) completion of jobs which require personal attention because of unusual difficulties or other special circumstances. It is also the responsibility of the User Support Group (together with other computer centre groups) to simplify the user's life by developing an easily accessible library of useful routines and packages to accomplish frequently encountered tasks.

Specific activities and services in this area include:

- the consultants' office(s)
- a "special handling" service for programs with unusual problems or requirements
- the development and maintenance of program libraries
III. Fault detection

It is the responsibility of the User Support Group to assist the other computer centre groups in the detection and analysis of (hardware and software) faults, both before and after installation of the product. This is accomplished by assisting in the development and administration of adequate pre-installation tests, and by careful analysis of anomalous behaviour reported by the users both during tests and during regular use of the product. It is also the responsibility of the User Support Group to verify refund claims.

The principal agency for this activity is the consultancy.

IV. To represent the users within the computer centre

It is the responsibility of the User Support Group to represent the interests of the users in internal computer centre discussions at the working level. (An advisory committee generally handles this responsibility at the policy level.) This involves in particular an examination of the user interface to proposed new elements of the system, the assessment of the user impact of proposed changes, the transmission of the user viewpoint with respect to the relative priorities of projects under consideration, and the transmission of suggestions from the users concerning what new facilities or improvements the users would like to see.

The principal agencies for the collection of user opinions are the visitation program, the suggestion procedure, and the consultants' office(s). The methods by which User Support carries these opinions into internal computer centre deliberations include participation in design and planning projects, formal and informal discussions with implementors, and (possibly) a leisurely exchange program between User Support and the other computer centre groups.
APPENDIX III

Improving the Interaction between User Support and other Computer Centre Groups

I. Management Commitment

In order to secure lasting improvement in any intergroup interaction it is first necessary to ensure that both groups understand both the necessity for such interaction and its purpose. Formal interaction can be instituted by fiat: productive interaction results only if those involved realise they are working towards a common goal, and that they will achieve it more easily through co-operation. Furthermore, workers reflect the true priorities of their management (which do not always agree with the published priorities). Thus, before any mechanical or organisational steps towards improved interaction are attempted, it must be made clear to all that the computer centre management accepts and endorses the philosophy which calls for such interaction. More specifically, it is necessary that management:

1. Explicitly and completely accept and promote the service orientation of the computer centre.

2. Explicitly and formally recognise the User Support Group as the guardian of the user interface to the system.

3. Explicitly and formally conclude that more intensive involvement is therefore required: by User Support in the design and planning stages of various projects, and by all implementors in the implications for the user of all their work.

4. Not sabotage its own explicit and formal statements by failing to support human engineering in the face of various pressures (including economic).

II. Kinds of Interaction; Standardisation?

Given that everyone now understands the mission of the computer centre, and User Support's portion of that mission, the range of the resulting intergroup interaction is seen to be quite broad:

1. The mutual exchange of plans.
2. The notification of bugs, design errors, and lacunae encountered.
3. The division of labour within an area of shared endeavour.
4. Mutual participation in the establishment of relative priorities of work in progress or about to be undertaken.
5. Joint concurrence on appropriate portions of design specifications (User Support must concur in the proposed user interface; Systems must concur in the proposed operating system interface, etc.)
6. The joint development (involving User Support for any new element which is not user-transparent) of suitable acceptance criteria.
7. The (verification of) passage of the acceptance criteria prior to installation.
APPENDIX III/Cont/d

In the absence of installation-standard policy and procedures, many of these discussions will degenerate into aggravating repetitions of earlier arguments. In order to prevent this -- or at least to reduce it to manageable proportions -- the installation should consider the standardisation of some of the more common interfaces. A few candidates are:

1. nature and timing of system changes and change notices;
2. criteria for accepting new modules into the system
   a) from within the computer centre
   b) from users;
3. user interface specifications for new modules;
4. acceptance criteria (including documentation) for new or changed modules;
5. system instrumentation and measurement;
6. newsletter policy.

III. Organisational Aids

Although it is people who solve problems, not organisations, sometimes the organisation can provide a little help. Two ways have already been mentioned: the acceptance by management of the idea that this sort of interaction will work to produce a better computer centre and the possible institution of standard policy in some areas. Another way is to try to provide for more physical mingling of the personnel of two (or more) groups whose work is closely interconnected. When the people mingle, their plans and priorities become common knowledge, their talents and ideas are pooled, their conflicts are aired; the desired interaction takes place on an informal basis. (If the interaction is confined to a formal procedure people often fall into an adversary rôle-playing trap.) Unfortunately, the design of the CERN computer centre buildings tends to make such mingling rather difficult: people are segregated into small cells, there are too many doors, and the corridors are not wide enough for conversation.

One formal analogue to physical mingling is a periodic exchange of personnel between two groups; another is the assignment of interested "outsiders" to each project, at least as observers. (For the purpose of this discussion, that tends to mean the appointment of a User Support person to each project with user impact, at least until the external specifications have been frozen.) A final, very important organisational aid is a method of arbitration. Disagreements will arise; they must not be allowed to blossom into antipathy (as they will do if allowed to fester). In those circumstances where the participants cannot agree, they must be able to appeal for, and obtain, a timely decision.
IV. Common courtesy

We are concerned here about the interfaces between several sets of people; in such interfaces how a thing is done is as important as what it is. It is therefore incumbent upon all computer centre personnel to remember the rules of common courtesy. A few of the more obvious commandments are included here:

1. Keep everyone informed of your plans and progress.
2. Be available for intergroup discussion on items of mutual interest.
3. In situations requiring joint agreement, get it before proceeding.
4. Fix the problem, not the blame.
5. Don't get personal: describe the fault, not the character and ancestry of the author.
6. Answer calls, messages, requests for information and assistance; even a negative answer is more courteous than none at all.
7. Be on time for meetings and appointments (including machine sessions and office hours).
8. Do your homework; failure to prepare for meetings is a good way to get your invitation cancelled.
9. Negotiation is better than arbitration, but arbitration is better than altercation.