Toward an inclusive information network

by R. R. HENCH and D. F. FOSTER

General Electric Company
Bethesda, Maryland

INTRODUCTION

In the next decade, an increasing proportion of computer power will be provided by information service networks, rather than by the tens of thousands of individual installations which now exist. This proposition is made attractive by many economies of scale, of which "Grosch's Law" is only one. Equally important are:

- The reduced cost of the redundancy which is required for high reliability.
- The drastically reduced cost of nationwide communications when provided in large quantities.
- Lower operations costs.
- Better capacity utilization due to more incremental growth.
- Variable, rather than fixed, costs for the user.

General Electric is committed to providing a viable service alternative to in-house processing for a wide variety of computing requirements. This paper describes the design of the systems currently under development which addresses that requirement.

DESIGN ALTERNATIVES

In developing a service for the entire DP community, there is an almost irresistible temptation to say, "Let's start from scratch and do it right." Under this philosophy (which we will call the *exclusive* alternative), the designer would develop a new and highly-generalized combination of hardware and software. Such a system aims at being all things to all men—it serves time-sharing and remote batch; the on-line user and the massive tape sort; the big LP problem and the inventory-control system. Unfortunately, this alternative ignores the existent multi-billion-dollar investment in software and knowledge for current systems. It also ignores the advantages which may follow from specialized hardware and software dedicated to specific kinds of applications or functions.

The other alternative (which we will refer to as the *inclusive* alternative) is to take advantage of pre-existing systems, integrating them into a Network but still maintaining each as an entity. Thus, advantage may be taken of the software and knowledge which exist for such systems, in addition to taking advantage of their unique capabilities.

The key problem, of course, is in providing meaningful integration among these systems. Many major application areas involve the interaction of diverse kinds of computing resources. For example, data may be collected on-line from geographically-distributed locations, then submitted for batch processing. If the on-line and batch functions are to be done on separate systems, a high degree of integration must clearly be provided between them.

DESIGN STRATEGY

GE has therefore undertaken the development of what we may refer to as a *Technology of Compatibility*, the aim of which is to permit the integration of diverse kinds of systems. The specific elements of this strategy are as follows (See Figure 1):

- Provide a generalized, widely deployed *Communications Network.*
- Provide an *On-line Service* of the highest possible quality and efficiency.
- Interface this foreground service with a variety of existent *Background Systems*, both GE-owned and (potentially) customer-owned.

Thus, the user will be provided with a common set of tools for on-line program development and for geo-

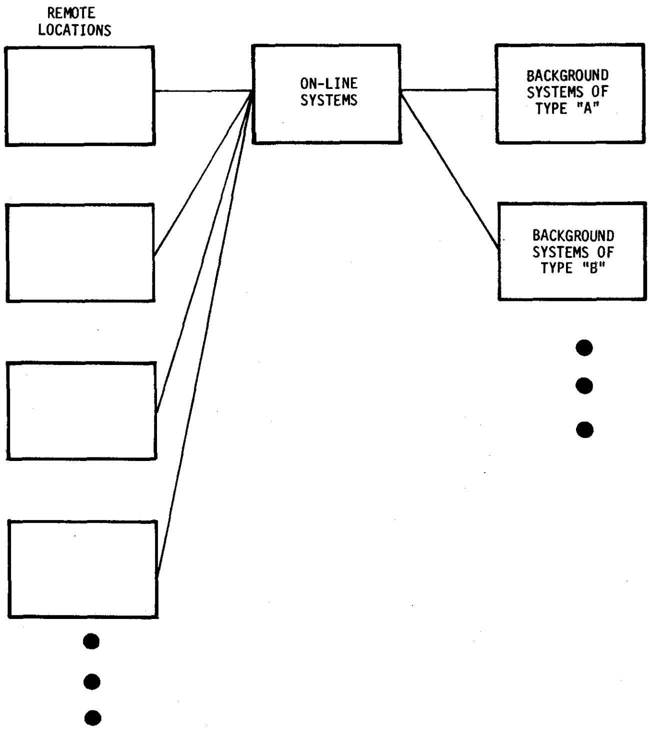

Figure 1—Conceptual design

graphically-dispersed data collection and distribution. At the same time, he will be provided with a choice of background systems, depending on his specific needs. Ongoing batch applications may thus continue to operate in a compatible environment while being provided with an efficient "front-end" for interactive processing.

The rest of this paper will be devoted to a discussion in some detail of the elements of this strategy—the communications network, the online service, and the foreground-background interface.

COMMUNICATIONS NETWORK

The Network extends to over 250 cities in North America and Europe. From each of these locations, the user may reach any of the central systems by typing an appropriate user number. The Network design permits the systems to be located physically anywhere within the Network.

Topologically, the Network is a polycentric star (Figure 2). The transmission and distribution paths throughout the entire Network are store and forward two-line logic with diversified routing wherever necessary. The Network is based on the use of distributed computers performing specialized functions.

Distribution—remote concentrators (Honeywell 416)

These computers are the outermost nodes of the grid. Their functions include:

- maintaining awareness of physical terminal characteristics (speed, character set, etc.)
- simple editing (line and character deletion)
- context recognition. RCs keep track of whether each terminal is entering *data* or *commands*. Thus, the central system will be interrupted only when commands (which require immediate central system action) are entered, and not every time the terminal user types a line of data.

Transmission—central concentrators (Gepac-4020)

CCs provide the interface between the central systems and the Network, and also between the RCs and the Network. Their functions include:

- buffering data to and from the terminals. A drum is provided for this purpose.
- determining to which central system a user is to be connected, at logon time.

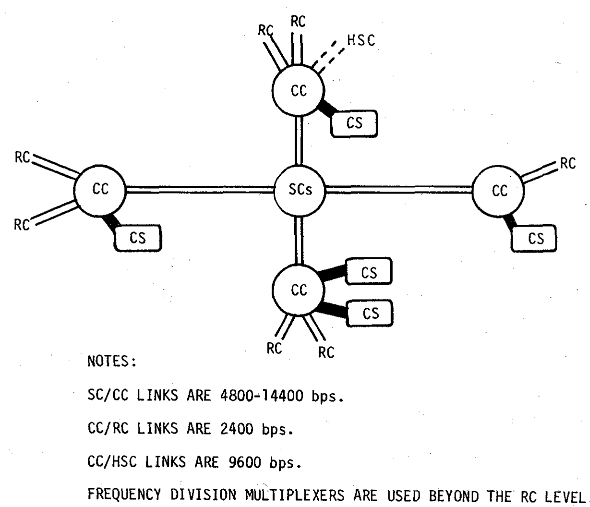

Figure 2—Network structure

- communicating with central systems, remote concentrators, and switching centers.
- maintaining awareness of Network configuration.

Switching—switching centers (Gepac-4020)

The switching centers provide a transparent link between Central Concentrators. At present there are two SCs, operating independently of each other logically as well as physically.

Processing—central processors (GE635, 605, 235, HISI 6000)

Central systems provide all user job processing. A number of central systems may be attached to each CC. The MARK IIs, accounting for most of the load, are connected directly via memory-to-memory interfaces. Any other type of central system may be connected with a modified interface (either hard-wired or remote) which uses a high-speed communications channel on the CC and a standard software interface discipline.

Network bandwidth extension

The Network has recently been extended to support synchronous terminals up to 4800 bits per second. This has been done consistent with the Network structure through the introduction of the high-speed concentrators (Diginet 1600s) interfacing with CCs much as remote concentrators do now. Message size and queuing philosophy are the only differences. HSC-CC communication will initially be via dual lines at 9600 bps. The HSCs will support a variety of terminal types, including computers. An OS teleprocessing package has been developed to permit 360 or 370 computers to communicate with the Network. Through a procedure called Interprocessing,® a user may transmit files between his in-house system and one of the Network's on-line systems. Thus, he may achieve nationwide access to his data bases for inquiry, updating, and further processing.

THE ON-LINE SERVICE (MARK II)

MARK II was designed and optimized specifically for interactive use. This includes conventional time-sharing, but more importantly the very different and demanding area of geographically-distributed, transaction-oriented processing. The hardware (GE-635) provides a master/slave concept and automatic hardware relocation of programs. The software provides the following general capabilities:

- a simple, user-oriented command system.
- task string logic. Each command is broken down into a string of standardized, elemental tasks. Tasks may schedule other tasks ahead of or behind themselves.
- heavy use of resident, reentrant code.
- a separate communications processor.
- a completely device-independent logical file system.
- multichannel swapping, permitting up to three swaps to be going on simultaneously.
- extensive security and integrity checking.

In addition, a number of capabilities oriented specifically toward the transaction-oriented data collection and distribution market have been provided. These include:

- file permissions, by means of which each user may exercise explicit control over access to his data base.
- file locking, permitting control over multiple-update situations.
- journalization, providing a common magnetic tape for the logging activities of multiple users.
- interprocess communication, whereby programs can signal other programs indicating processing tasks for them to perform.
- user control. The activities of a terminal user can be placed under complete program control. When the user logs on, a specified program may be automatically invoked, and escape may be prohibited. Almost all system functions may be done by means of CALLS from a high-level language program, eliminating the necessity for the eventual terminal user to give system commands.

THE BACKGROUND INTERFACE—MARK III

As discussed earlier, it is our intention to interface MARK II with a variety of background systems. The first step in this direction is MARK III, which provides an interface between MARK II and the GECOS-III remote-batch system. Although GECOS runs on hardware similar to MARK II, it is a completely different operating system, optimized for batch and remote job entry to the same extent that MARK II is optimized for on-line activity. The interface permits each system to specialize in the kind of work at which it is best:

- all on-line work is handled by MARK II
- all background work is done by GECOS
- separate file systems are maintained
- a high-bandwidth interface is provided. All background work is submitted by MARK II across the interface.

Design principles

Although the specific code used to interface MARK II with GECOS-III is probably not transferable in interfacing MARK II with other systems, the fundamental design concepts most certainly are. These are as follows:

- Changes to the MARK II and GECOS-III operating systems must be kept to a minimum. Absolutely no changes could be made which would in any way impact current users of MARK II or which endanger compatibility with future manufacturers' releases of GECOS.
- The foreground and background systems must be independent of one another. The failure or intentional shutdown of one must not affect the other.
- The interface or mailbox protocol must be adaptable to communications technology and not involve physical constraints such as distance or storage.
- It must be possible to interface multiple MARK II systems to a single GECOS-III system.
- It must be possible to transfer very large files and output reports between the systems efficiently.
- Character and file type conversions must be handled as straightforwardly and as automatically as possible.
- It must be possible to use the background system conveniently from either a high-speed or low-speed terminal.

Physical interface

The initial linkage between the MARK II and GECOS-III systems is a shared disc device (Figure 3). This is *not* in any sense a shared file system; it is simply used as a high-bandwidth communications channel. The transmission of messages between the two systems is completely asynchronous, using what is called a "passive mailbox scheme." One disc mailbox area is written only by MARK II and read by GECOS-III. Another area plays the converse role. All interface

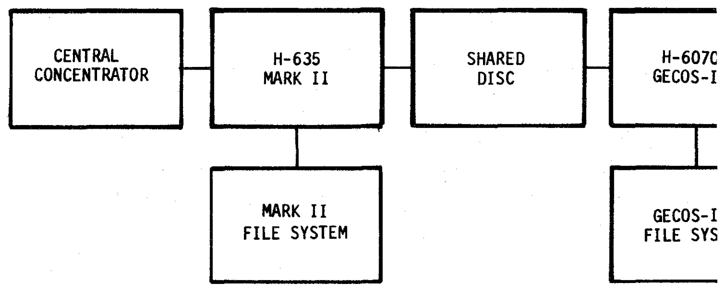

Figure 3—MARK III

status information is maintained on the disc itself, to facilitate restart in the event of the failure of one of the systems. It should be understood that the use of the disc is merely a practical solution and plays no active role in the logical interface discipline.

Logical interface

A background job may originate from either a high-speed or a low-speed terminal. In either case, the request will be generated on MARK II and will be transmitted across the interface to GECOS. Background job initiation is via the BACK command. BACK requires a set of *directives* describing both the interface work and the background work to be done. At present these directives include:

- run a job file, i.e., place it in the background system input stream.
- transfer given files between foreground and background, with appropriate file format and character set conversions.
- create and purge background files.

Standard system job output (SYSOUT) is automatically returned to the foreground system and placed in a special library. At the user's option it may be:

- directed to a high-speed terminal.
- scanned by a low-speed terminal, using special editing features.
- placed in the user's permanent file system.

Conversion disciplines

A variety of character sets coexist in MARK III. ASCII is primarily used by the foreground system and by low-speed terminals; BCD is used by the background system. Both ASCII and EBCDIC are used for communicating with high-speed terminals. This proliferation of character sets is a fact of life for anyone who aspires to serve today's and tomorrow's information society. In order to avoid chaos in dealing with it, it was necessary to establish a system-wide discipline for dealing with conversion problems.

All files stored on MARK II have their character set content (when meaningful) retained in the catalog. Processes which must perform conversions will utilize this information. Conversion processes will retain identity of logical records, and will also maintain special control information, such as printer slew control characters. The conversion is thus considerably more complicated than a simple one-for-one transliteration. By default, all data files transmitted are converted from the natural conventions of the host machine to those of the target machine. SYSOUT files are, however, maintained in original format and with the same record structure which exists on the background system. This is done for several reasons:

- to avoid unnecessary conversion. Much background output is simply scanned, using the low-speed terminal editing procedures; then thrown away.
- to save file system space (6 bit BCD characters as opposed to 9 bit internal ASCII representation).
- to avoid double conversion. If output is eventually destined to a high-speed terminal, for example, its character set will probably be EBCDIC. The user, however, may well not decide whether or not he wishes to print his output on a high-speed terminal until he has scanned it.

Software implementation

- BATCHER, a MARK II slave software module, is invoked whenever the BACKGROUND command is given. It checks validity of the directives and queues the job for transmission.
- INPUT MONITOR is a continuously-running slave module which takes jobs from the queue and writes them to the shared disc. At this point all files which are to be taken over to the background are merged in.
- MARK2 is a GECOS privileged-slave module which handles all communications with MARK II via the shared disc. It is written in reentrant code and can communicate with multiple MARK II systems simultaneously.
- PREP (preprocessor) is a job spawned for each user ahead of his requested background work. It handles conversions, file creations, etc.
- OUTPUT MONITOR is a MARK II continuously-running slave module which handles all output returning from GECOS across the interface. A variety of other status and editing modules are also provided.

It is significant to note that the actual changes to the background system have been minimal. This validates the design philosophy of truly connecting alien operating systems without compromising software maintenance by the hardware vendor.

Applications programs

GE has always made heavy use of applications programs to provide higher-level interfaces to its systems. This policy will be continued with MARK III. Several kinds of application programs which operate across the interface are being developed:

- control card generators. These programs will interactively query a user and will generate for him the control cards necessary to perform most basic background functions.
- application initiators. These will exist in connection with a specific background application program (for example, linear programming). In addition to generating the control cards, they will perform validity checking on the user's input data and spawn the optimum job relative to resources needed.
- distributed applications. These are application packages which make use of both the foreground and the background, applying each to the tasks it is best at. For example, an accounts receivable package might maintain customer credit level information in the foreground, but keep customer name and address records, purchase records, etc., on tape in the background.

THE FUTURE

GE's developments in the Information Services Business have been based very largely on distributed intelligence techniques. Future work will largely consist of extensions of these concepts. Some significant future developments are discussed below.

Multiple background systems

As indicated earlier, the evolution of the MARK III concept to various and diverse background systems is planned. As more systems are integrated into the total service, there will follow closely the natural extension of remotely-located backgrounds. With time the interface will permit background systems owned and operated by customers to become an integral part of the product. The implications are obviously very penetrating. It will be possible to procure a highly reliable interactive foreground service capable of meeting all the demands of continuous on-line services, which in turn is coupled with one's own in-house machine. From the same terminal one could cause activities to be spawned back into one's own system.

Shared second-level file system

In any multicomputer configuration it becomes increasingly undesirable to require manual intervention for handling of such removable media as tapes. For this reason, we are closely following the evolution of massive storage device technology. When this technology becomes sufficiently mature, we will implement such a device, shared among all systems in a given location and provided with intelligent controllers. This will create a hierarchical storage structure in which the lower level of the hierarchy is common to many systems. Information will move from one level to another at the explicit or implicit request of users.

Communication of complex data structures

Initially, file transfer across the interface has been limited to character-oriented sequential files, since these are common to all operating systems. It is desirable to provide transfer of more complex file types, including random structures and binary data. Clearly the problems involved in converting such files from the conventions of one operating system and set of hardware to those of another are non-trivial, but it is possible that a reasonable solution can be obtained for some substantial subset of all files.

Multiple foreground systems

At present, MARK II provides a high degree of reliability through the use of a redundant "swing system". If a system fails, its users may be transferred to the swing system by logic in the communications processors. Peripheral switches move files logically from any seriously crippled system to an idling reserve. This does not address the interruption problems of on-line services but has made a tremendous stride in availability.

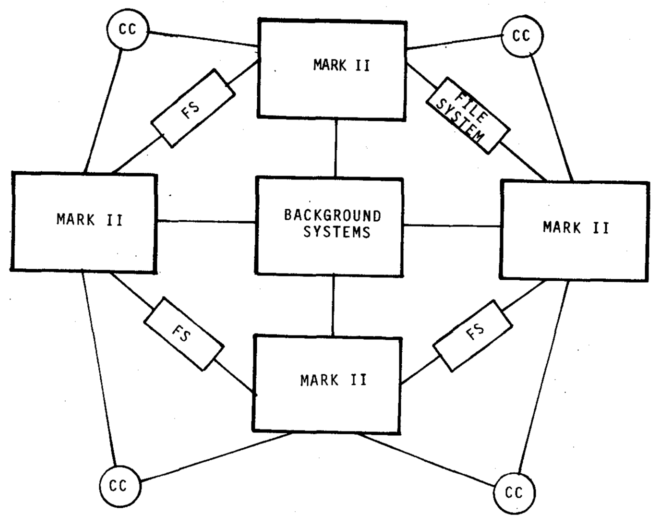

Figure 4—Ring system

The next major step to extend the foreground reliability will be changing the current processor focus to a file focus. The current architecture is so structured that the system topology is a function of processors. This must change to make the pivotal nodes the file subsystems. This will be done by having the file and processor subsystem interlaced in a concentric circle as shown in Figure 4. Users will be assigned to central systems dynamically by the communications processors. If a system fails or is taken off-line, its load will immediately be taken up by the two adjacent systems. In a short time the load of the failed system will be distributed evenly across the ring. A stable ring of n systems reduced to $n-1$ will re-stabilize within 30 minutes due to normal log-ons and log-offs.

In addition to its reliability advantages, this configuration provides significant economies by balancing load fluctuations across systems. Preliminary analysis indicates that these economies may exceed 15 percent of total capacity.

COMMENTS

The Network described here has grown in a somewhat organic manner, subject at all times to the hard discipline of commercial viability. It is not our intent to lay down an inflexible master plan, but rather to develop a general structure with the ability to adapt to changing conditions. Inevitably, some aspects will be developed more intensively than others as experience indicates their value. The character and scope of those features which are commercially implemented are therefore subject to alteration. Hopefully, the structure described will provide sufficient flexibility to accommodate this continuing redesign and to provide for the smooth integration of new technologies into the ongoing operation.

ACKNOWLEDGMENT

The ongoing developments discussed here are implemented under very tight schedules and by a surprisingly small group of people. The authors would like to express our appreciation to our colleagues since this has been a group effort from its inception.

REFERENCES

1 *On line time share computer services—A telecommunications industry survey*
TELECOMMUNICATIONS Volume Four Number Eleven November 1970

2 G E GAINES JR J M TAPLIN
The emergence of national networks—Remote computing—Year VI
TELECOMMUNICATIONS Volume Five Number 12 December 1972

3 R C HAAVIND
The three phases of shared computing
COMPUTER DECISIONS November 1971

4 G J FEENEY
Information network technology
General Electric Company Document No 238010 2/70 Presented at the Congres International d'Informatique Paris September 23 1969

5 J C CASTLE
Communication system design in multiple access computer networks
Memorias de la Conferencia Internacional IEEE Mexico 1971 Sobre Sistemas Redes y Computadoras January 19–21 1971 pp 636-640

6 F D MONTGOMERY JR
Computer resource sharing through communications network technology
EUROCON 71 Digest IEEE Catalog No 71 C 56—REG 8 October 18-22 1971 Lausanne Switzerland

7 W S HOBGOOD
Evaluation of an interactive-batch system network
IBM Systems Journal Volume II No 1 1972

8 L G ROBERTS B D WESSLER
Computer network development to achieve resource sharing
SJCC 1970