Automatic Calling with the HP3000

Paul W. Ridgway
Systems Engineer
Hewlett-Packard Company
Houston, Texas

INTRODUCTORY SUMMARY

The purpose of this paper is to introduce HP3000 users to the concepts and benefits of computer controlled call origination. Accurate information management is one of the foundations for a successful business and as businesses grow geographically, accurate information management between businesses becomes increasingly important.

The topic of information management between businesses has been addressed through the communications network concept with each vendor supplying their own competitive network design. In this paper, I will discuss methods of automating the existing communications network capabilities of the HP3000 and will propose new ideas of automating the communication of electronic information.

The body of the paper will consist of two sections. The first will be a technical discussion aimed at the non-technical individual. It will describe the components involved and how they interface the HP3000 with the outside world. Also, the first section will examine the software required and will give examples of how to integrate the software into existing network modules. The second section will deal with the concepts of automatic call origination and will include a discussion on the need for automatic information transfer and a summary of existing and proposed appplications which are suited to the HP3000.

To conclude, we will work out the economics of automatic call origination and compare its advantages to other methods of networking.

INTRODUCTION

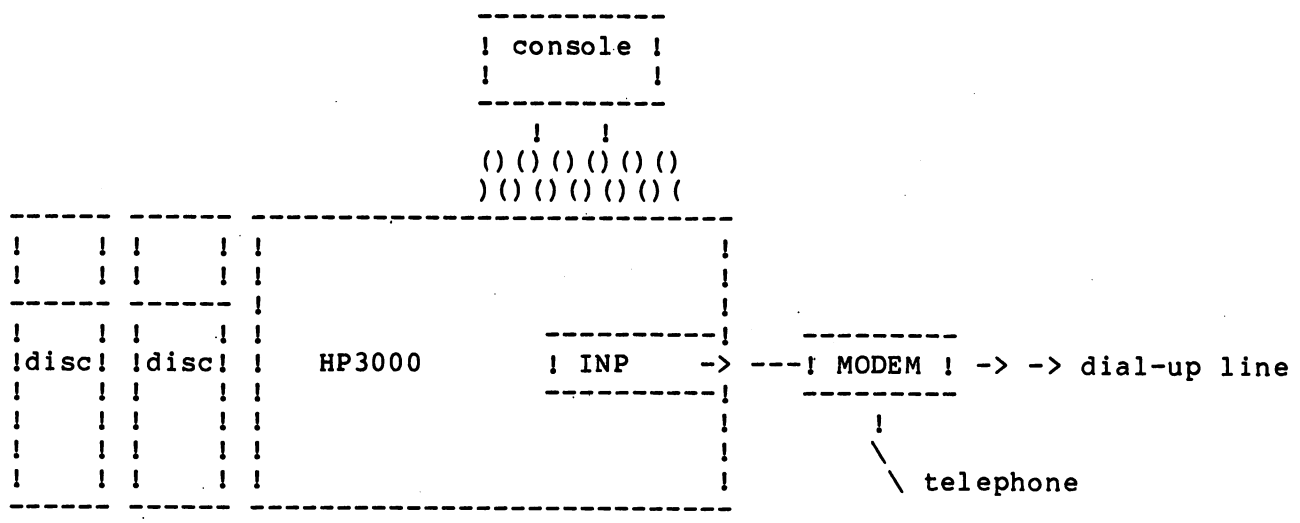
I am an observer of the Cosmos to use Dr. Carl Sagans' word to mean that I get input for my thinking from every source of information known to man. For example, if I want information about the universe, I select a source of that information, say a book, and read. If I want information about the state of politics overseas, I select a television channel that is displaying that information and watch. In these cases, I know what *kind* of information I want, I know *where* to get it and I have the ability to make contact with the source of information through my actions.

This seeking out, absorbing and updating of data is called information management and is an integral part of human existance as well as of modern business. For example, a company with employees that are paid weekly has a data entry method for giving the computer information about the number of hours worked, sick leave and vacation earned or taken, pay rate increases and so forth. Once each Friday, the payroll program runs, extracts the information and produces output in the form of paychecks. For a single office business with few employees, (and small quantities of information) this scheme works fine. However, if the business has a home office and several branch offices across the continent, the communication of information becomes a formidable and costly overhead operation.

This paper will address two facets of information communication: the ability to select the information source or destination and the timliness of communication with regard to cost effectiveness. To clarify these facets: automatic selection of source or destination is analogous to me as an observer; I can select a book or television as the source, depending on what I need to know. The timliness of communication is analogous to the idea that I can select the source or destination when I need it and in the case of the home office — branch office example, when it costs less to transmit or receive the information.

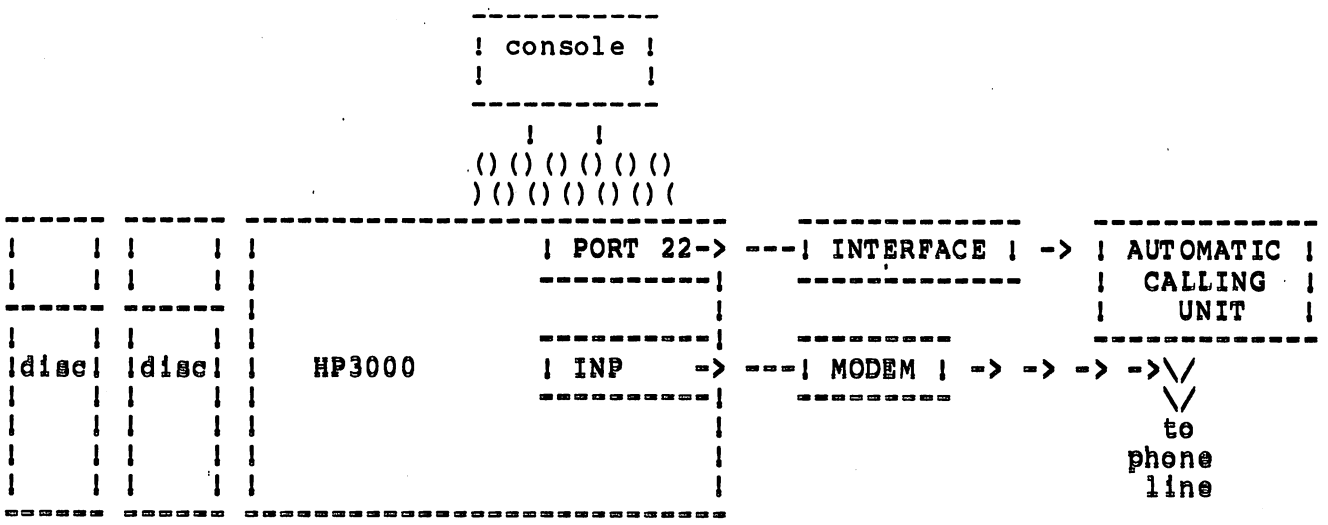
HARDWARE and SOFTWARE

Let's now take a look at the first part of the discussion, the autodialer hardware. Consider this typical environment:



Using HP's Distributed Systems software and the Intelligent Network Processor (INP), a user of this system can manually place a call to another HP3000 of similar configuration having auto-answer capability and transfer data in either direction by issuing the right commands. Neither system has any special capability and call origination is done manually by the operator.

To apply an autodialer to the system, the diagram would look like this:



The additional hardware is the Autodialer Interface and the Automatic Calling Unit (ACU). Within the ACU is a special mechanical switch which determines whether it or the modem is connected to the dial-up line. This switch has two states: "ready" and "busy." When the line is not being used the switch is in the "ready" state and connects the phone line to the ACU. When a communications link has been established with another system, the switch is in the "busy" state and the modem is connected to the line.

The effect of the autodialer (consisting of the interface and ACU) is to remove the manual operation of picking up the phone, dialing and waiting for a modem to answer and placing the data set online. Control and monitoring of the autodialer is done entirely through software which sends and receives control codes through the asynchronous port 22. Note that the actual data exchange between the two computers has not changed; existing DS software is used as before.

The ACU is the workhorse in that it is responsible for siezing the line, acknowledging a valid dial tone, sending a sequence of tone pairs down the line (the phone number) and recognizing the response. Typical responses from dialing a number are: no response, no answer, busy signal, human voice answers or a valid modem answers. In the case of a modem answering the line, the ACU can tell what type of modem responded by the carrier frequency placed on the line by the modem. The ACUs job is to tell the interface what response occurred. If a modem of the right type answers, the ACU must also flip its internal switch transferring

control to the local modem. The interface and ACU communicate with each other using a protocol called RS-366, the paralell equivalent to the familiar serial RS-232C standard. Thus, the interfaces' job is to translate RS-366 to RS-232C ASCII and vice versa such that the HP3000 can control the ACU. The HP3000 software does this by sending a string of ASCII characters to the port and reading the port for the response as though it were a simple file.

For a look now at the software side, this is a skeletal flowchart for a procedure that would connect any two systems together:

```
PROCEDURE DIAL ( PHONE'NUMBER , RETURN'CODE );

        START
          !
          !
    READ INTERFACE STATUS
          !
     IS LINE OCCUPIED?  YES -> RETURN'CODE := 1 -> RETURN;    (ERROR)
          ! NO
          !
     SEND A PHONE NUMBER
          !
          !
    READ INTERFACE STATUS
          !
       STATUS = "A" ?    YES -> RETURN'CODE := 0 -> RETURN;    (NO ERROR)
          !                                                   (MODEM ANSWERED)
          !
       STATUS = "B" ?    YES -> RETURN;CODE := 2 -> RETURN;    (ERROR)
          !                                                   (BUSY NUMBER)
          !
       STATUS = "C" ?    YES -> RETURN'CODE := 3 -> RETURN;    (ERROR)
          !                                                (INCOMPLETE DIAL)
          !
HAS 30 SECONDS ELAPSED?  YES -> RETURN'CODE := 4 -> RETURN;    (ERROR)
          !                                                   (NO RESPONSE)
          !
    END PROCEDURE DIAL
```

A program containing higher level software would see the autodial process as a simple procedure or intrinsic call to the system and would pass and receive data through a parameter list. In the case of any error response, the calling program would have to decide what to do next; possibly retry the call or give the user a choice of actions. If the "A"' response is given, the program would proceed by logging on the remote system and commencing a data transfer operation. The flowchart for a "driver" program would appear as follows:

```
      PROGRAM 'DRIVER'

          START
            !
            !
      DIAL REMOTE MODEM
            !
            !
GET A LOGICAL DS LINE NUMBER
            !
            !
        LOG ON THE
      REMOTE SYSTEM
            !
            !
```

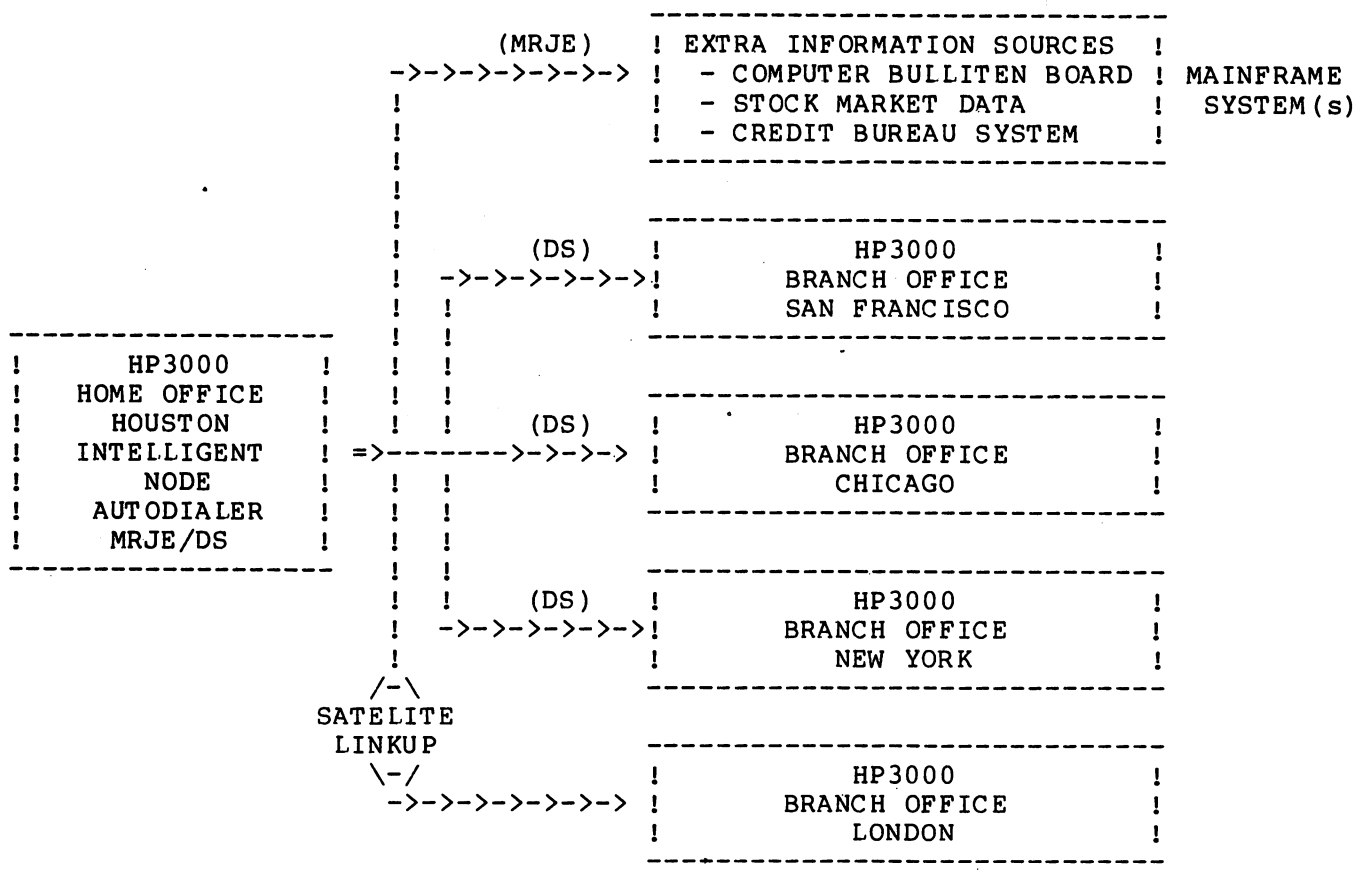
```
    SCAN THE 'AGENDA'
      FILE ON BOTH
        SYSTEMS
          !
          !
    PERFORM INDICATED
     DATA TRANSFERS
          !
          !
      LOG OFF THE
     REMOTE SYSTEM
          !
          !
CLOSE THE DS LINE
          !
          !
END PROGRAM 'DRIVER'
```

Within the driver program, the actions of scanning the agenda files and performing the indicated operations steps would have been previously set up according to the application using the autodialer.

CONCEPTS

This part of the discussion will deal with the concepts of automatic calling. The autodialer may be viewed as a special "hook" into a network of computers. The term "network" can apply to both existing connections between systems as well as *possible* connections between systems. In any case, the ability to share and thus manage electronic information lies in networking theory. To illustrate, let me present a quote from a popular small systems journal: "Some people forsee electronic information as the currency of the future: those who have it will use it to get more and those who don't have it will be exploited. Actually, money will probably continue to be the currency for years to come, but the computer will be the primary tool for controlling its flow. The key to this flow lies in computer networks. With the price of individual computers dropping, more businesses are solving their problems with distributed processing of computer networks rather than with a single large computer."[1]

If we accept this concept of dependence on the network, automatic access to the network is certainly a step in the positive direction. Giving a single system the capability of automatic access creates what I will call an 'Intelligent Node', one that can make unattended decisions based on access to the network. The intelligent node is actually a "smart" piece of software consisting of several logical modules, each having a specific responsibility. One such module is the autodial process. Another would be the one that logs on the remote system and another the one that handles file transfer and so on. A special feature of the intelligent node is that it is invisible to the network process. As an example, consider this diagram of a hypothetical warehousing and distribution business:



In this drawing, a business with distributed data processing is outlined. The home office is the "hub" or center of the network and has an INP configured for Distributed Systems and Multileaving Remote Job Entry use and an autodialer module. Surrounding the home office are the four branch offices each having a smaller HP3000 with DS and autoanswer modems. The London office is contacted through a commercially available satellite link such as Telnet and is accessed by the home office via the normal dial-up method.

Consider the following scenario:

Business function: Sales, warehousing and distribution of an industrial product or service which varies in cost, supply and demand on a daily basis.

Sales and product stocking are carried out at the branch offices while administrative and purchasing operations are handled by the home office. Data for pricing and sales volume need to be sent to and received from the branch offices on a daily basis to maintain the competitive edge. General electronic information about the stock market and credit information on the company's customers is stored on one or more systems which are not owned by the company but are used as extra information sources. The cost of the extra information depends on the demand based on the time of day; daytime access costs more than nighttime access.

Problem: The company is paying for an extra system operator whose only function is to manage the daily task of establishing contact with the remote systems and transferring the data to and from them. The cost of data transfer is high since all transactions take place between 8:00 am and 5:00 pm when the telephone rates are highest. The extra information sources charge high rates since access to them is made during prime time hours. The process is error prone since the operator follows a changing schedule which is often inaccurate.

Solution: The installation of an autodialer in the home office system removes the task of manually establishing contact with the remotes thus allowing the operator to perform a more valuable function for the company. Since all the systems are running 24 hours a day, data communications take place at night when phone rates are lowest and the extra information systems cost less to access. The process of contacting each system and performing the data exchange operations is done through a software "script"; a file of instructions for the intelligent node software to follow. Since the human element of error is removed, more accurate data transfers take place and the phone line use is optimized.

Operation: Every night beginning at 10:00 pm the Houston office automatically dials and connects with each remote system. The sales pricing and stocking data are exchanged with the remote systems such that when the next days work begins, every office has recent and accurate data. The extra information systems may or may not be contacted nightly depending on the need for that information. For example, if an employee in the London office wants credit information about an American client, she/he would put a request for that information in an "agenda" file on the London system. When Housto calls London that night, its software would scan the agenda file on the London system and receive the request for credit data. Then, the Houston system would contact the credit data system and extract the necessary information. When that transaction is complete, Houston would re-dial London and transmit the file to disc via DS software. Also, the software would make an entry to a log file documenting the exchange. The next day (or afternoon) the London employee would look at the log file, note that the data tranfer had taken place and find the name of the disc file containing the information she/he needs.

Results: The company has made more valuable use of an employee by removing a remedial task and assigning her/him a greater responsibility. The cost of communication has been reduced through the advantage of lower long distance rates at night. The extra information cost is also reduced since the non-owned systems are contacted during non-prime time hours. The company has increased its competitive edge since the remote systems always have accurate and up to date data concerning sales and stock.

The operational procedure of the intelligent node software running on the Houston system is the heart of the entire process and would be capable of dealing with any predefined situation that could occur. For instance, if any system being contacted did not respond or if a data transfer error occurred, the software would "know" what to do: warn the system manager, retry the transmission or attempt some corrective action.

Well, so much for a hypothetical example. Now I will describe two existing applications which use autodialers. The first is an HP3000 user who sells time on their system to clients for general business accounting use. One particular client is a local business that is a branch of a company that uses very large mainframe computer systems. The local branch office has only a terminal and communicates with the HP3000 with a 1200 baud acoustic coupler. In operation, the client maintains an IMAGE database containing data about the local branch activity. On an as-needed basis, the client will run a program which uses the autodialer to establish a dial-up link between the HP3000 and one of three mainframes located in distant cities. Then the MRJE/3000 subsystem is used to transmit the database to the host mainframe. Since the MRJE software is designed to automatically submit pending files and JCL whenever a data link is established, the process is very adaptable to the control of an autodialer. Without the autodial capability, the client would be limited as to when the mainframes could be accessed since an operator would have to be on duty at the HP3000 and would have to be trained in the operation of the dial-up link. This additional overhead would be an extra expense for the HP3000 owner and would be passed on to the client. Since the client is a small branch office, this additional, recurring cost would make the entire operation unfeasible.

The second example is an in-house application within Hewlett-Packard. This system uses autodial to communicate with other HP offices as well as with supported accounts having the DS capability. Software was written to allow a user of an in-house system to establish contact with other HP3000s on a named basis. For example, when I need to get a file from a system at our factory in Cupertino, I log onto my system and enter a single command with a parameter that is the "name" of the system desired. The command is actually a UDC which runs the software.

The program then finds the specified name in a database nd extracts the phone number of the remote modem and checks the user's security code. If the user attempting contact is not authorized to use the autodialer, he is denied access and an entry of the illegal attempt is logged to the database. This is a security measure designed to protect the remote systems by never allowing the users to know the phone numbers stored in the database. If the user is authorized to connect with the remote, the autodialer places the call and the link is established. Finally, the software gets a logical DS line number by calling the COMMAND intrinsic and prompts the user for a valid HELLO command. When the remote HELLO operation is successful, control is returned to the user. In all cases, the software logs the operation in the database, allowing us to measure the use of the communication facility. In the event of a user requesting access to an unknown system name, the program enters an heuristic mode where the user is prompted for the new phone number and a security code which defines what other HP3000 users may call the new remote system. The new name, number and security code are then verified for accuracy and entered in the database for future use.

CONCLUSIONS

As has been shown, the autodial hardware is capable of giving users of the HP3000 quick, easy and accurate access to other sources of information. With the cost of communication on the rise and the ability to get and manage electronic information becoming more crucial to todays business, the autodialer certainly has a niche in the data processing field.

One topic of concern is the economics of autodialing. How much does it cost? In the earlier days, about 5 years ago, Bell Laboratories introduced the model 801 automatic calling unit and leased it to qualified accounts for a monthly fee between $30 and $60. A Bell modem was also required and cost anywhere between $100 and $400 per month plus a healthy installation charge. In more recent times, with Bell removing the direct connection restriction to their phone lines, several vendors have introduced their own versions of datacomm equipment and Bell equivalent modems. One such vendor of autodial equipment is Racal-Vadic, offering an ACU for a one-time charge of about $750. Their ACU works with any modem they make and is ready for direct connection to the HP3000 and the phone line. Furthur, one ACU may be instructed to do the dialing for up to 16 modems of different types. With this ability, a local HP3000 using autodial to connect to the console port of a remote system through an asynchronous modem could in fact control the remote system completely - from performing file backup to actually shutting the system down and restarting it.

The concepts presented here are not new to the mainframe systems; they are new to the mini-computer users. This is due to recent quantum leaps in communications technology for minis and the increasing importance of electronic information management between businesses who use minicomputers.

If having access to information is the lock to a successul business, then teaming computer systems with autodialers is the key to that lock.

BIBLIOGRAPHY

[1]Peter B. Reintjes, "Network Tools, Ideas for Intelligent Network Software," BYTE magazine, vol. 6 no. 10 October, 1981, pg. 140