INTEGRATION OF THE KERN DSR-11 ANALYTICAL PLOTTER INTO A GIS

By

W. Faig and J. Olaleye
Department of Surveying Engineering
University of New Brunswick
Box 4400
Fredericton, N.B.,
Canada
E3B 5A3

COMMISSION II

ABSTRACT

The key to unlocking the possibilities of a GIS is a good and up-to-date database which is provided primarily by surveying and mapping methodologies. Todays analytical plotters are typical state-of-the-art mapping workstations capable of providing accurate 3-D spatial information of large areas of the earth. As users of maps and charts are taking advantage of computer technology in their own endeavours and are requesting terrain information in digital form, mapping organizations are acquiring commercial software tools such as CAD and DBMS packages or developing their own systems in-house in order to meet the requirements of their customers. However, while the use of such tools represents an acceptable way to present survey measurements, the full potential of the analytical plotter is realized most dramatically when operated as a workstation in an integrated spatial information environment.

This paper describes the procedure used to achieve the integration of the Kern DSR-11 analytical plotter into a GIS environment. The issues of data format incompatibility between the source module and the GIS, isolated operation of the source module or the GIS, and limitations of the computing device are discussed as well as the problems they pose for successful integration. The operational mechanisms of both the system and the user interfaces are described. It is concluded that direct integration of the photogrammetric workstation to a GIS has many advantages which will increase with improvements in digital technology. However, the problems of data structure incompatibility, "closed" operation of the GIS package and limitations of the available computer facilities are obstacles in the way of realization of such integrated systems.

KEY WORDS: Analytical Plotter, Workstation, Integration, Child, Mailbox, Parent, GIS

1 INTRODUCTION

Rapid developments in both computer hardware and software technologies have had a remarkable impact not only on the way maps are made by photogrammetric and cartographic methodologies, but also on the way they are used to solve a variety of environmental problems.

Technically, today's analytical plotters are state-of-the-art mapping workstations capable of providing accurate 3-D spatial information of large areas of the earth, with varying levels of automation. Moreover, the almost exponential growth of digital technology has continued to facilitate the use of digital images in the mapping process, thus providing tremendous operational flexibility and reduced cost.

Developing in parallel, cartographers have abandoned the pen and paper technology and have adopted digital methods, making use of electronic pulses and files. The tedium of map construction by scribing is over; now, symbolization and editing of graphic elements are done electronically; map elements can be selectively displayed on graphic screens with great visual appeal and clarity. More importantly, the realization of the ability of the computer to store, retrieve, display, and even to use maps, has lead to the evolution of the modern digital technique of processing map information, thus, creating the all-pervading "Geographic Information Systems" (GIS) technology.

From the consumers' viewpoint, the use of map information has witnessed significant improvement due to the development of digital information processing which has immensely increased the value of the map as a decision-making tool. The ready availability of the computer, its reduced cost, increased power and ease of use, coupled with its ability to extract geographic information from digital maps with speed and precision, and even to analyze this information to propose solutions to problems has caused users to demand maps in digital form. Nowadays, spatial information is being used for large scale data analysis to solve problems in diverse areas, such as environmental monitoring, regional development, land use planning, facilities management and construction, exploration and management of the earth's resources. Evidently, computer technology has not only improved the production of the map, but also improved its use, thereby increasing the demand for spatial databases.

In general, one of two approaches may be used to produce digital databases for GIS applications from a photogrammetric workstation, namely (1) standalone operation in which the analytical plotter is linked to CAD/DBMS software modules (2) integrated operation in which the plotter is used as a workstation in a GIS environment. While the use of CAD/DBMS modules is an acceptable method to provide the needed data, the integrated approach offers a number of advantages. These include (1) immediate availability of data for GIS/LIS applications (2) possibility to overlay data from the application on the original image at the plotter workstation for visual comparison, interpretation and further analysis, i.e. the plotter becomes a GIS application tool (3) elimination of the need for data interpreters (4) cheaper production cost (5) flexibility of updating the database. Evidently, the integrated approach is highly desirable.

In developing an integrated operational environment involving components from different vendors, a number of technical problems prevent the full realization of this goal in most organizations. These problems include (1) data format disparity between the mapping workstation and the GIS, (2) isolated or solitary operation of one or more software packages, and (3) limitations of the computing facilities. The first problem makes communication between the units difficult or impossible; the second renders real time communication impossible, while the third may limit the capability of the resulting system or make its implementation impossible. These problems were solved for the integration of the DSR-11 workstation to the CARIS GIS at UNB, using the multitasking facilities provided by the VAX VMS operating system and the server interface module of the CARIS GIS.

2 TECHNICAL PROBLEMS INVOLVED IN SYSTEM INTEGRATION

2.1 Data format disparity

The process of system integration involves the establishment of an inter-process communication channel (system interface) for sharing and exchange of information between systems performing related operations. In practice, it seldom happens that both the photogrammetric system and the GIS are from the same manufacturer. Thus, frequently, attempts to integrate them are faced with the problem of incompatible data structures. Consequently, a greater per centage of the integration effort is spent on developing data interpreters between such components [Hodgson et al 1989, Ramirez 1989, Olaleye 1992].

Practically, the effect of data disparity depends on the type of integration desired - serial operation or parallel operation. From a technical viewpoint, integration for serial operation is a relatively simple process since the operations of the units involved are separated in time; therefore, turnaround time is often not a critical factor in the communication process (i.e. may be hours, days or even months). Consequently, the required communication interfaces may be as simple or complex as would be necessary to transfer information from one node to another node.

Firstly, it must understand and translate the data source into the form understandable to the GIS. Secondly, it must understand and be able to translate data and commands from the GIS to the source unit. And still, the translation has to be done in real time, implying that an active interface consisting of two translators, and able to operate within a short time cycle is inevitable. Of course, the translators must know the two opposing data structures in their entirety, otherwise they cannot achieve an error-free translation.

Thus, to achieve system integration in the face of data structure differences, two solutions may be considered: (1) employ intermediate data translators, (2) replace one of the software components with another having the right data structure. Logically, the two solutions are expensive but obviously the first, in addition, may lead to a complex and an inefficient system for real time applications. The second solution, though initially expensive may pay off in the long run. With the facilities provided for user programming in our GIS (as discussed below), the second solution was adopted in our integrated system.

2.2 Isolated operation

This problem is created when a component involved in the integration is incapable of communication with an external program, i.e. it denies access to its datafiles while in operation, for example, when a GIS package neither provides for nor accepts real time communication with an external program. Even if they have the same data structure, real time communication is not possible. Therefore, apart from data incompatibility, "closed" operation of individual programs is also a problem when assembling systems for simultaneous operation.

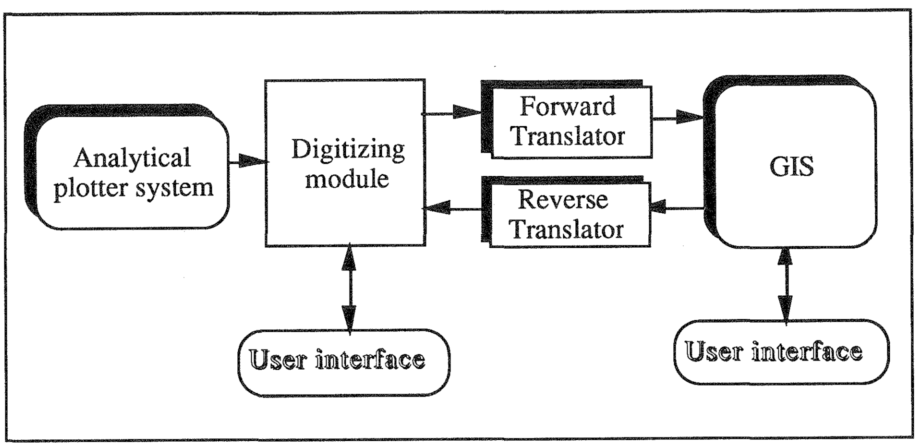

Figure 1: Integration of components of different data structures

On the other hand, integration for parallel operation involves synchronizing the operations of the system units so that they share information in real time. In this case, the situation is compounded by the fact that response time is a critical factor in the process. The rate of information transfer must match the pace of operation, particularly in an interactive operation involving the human operator. If the two units involved in the parallel integration possess compatible data structures, a passive interface which is consistent with real time operation provides the logical choice. For example, for systems operating in the same environment such as digitizing with real time editing, such a passive interface could well be a data record (or a datablock for a large volume of data) which is accessible to both system units for writing into and reading from. Yet, it is important that the writing and reading operations by the cooperating programs be coordinated to avoid collisions. If however, the components have data structures which are mutually discordant, then the problem of parallel integration becomes convoluted in that the interface module must perform two translations to achieve the desired link (Fig. 1).

Nonetheless, a recent and popular trend is that most advanced GIS packages such as the CARIS GIS now provide programming interfaces to which an external program may be connected for synchronized operation. Complementarily, most of the modern operating systems (for both mini- and micro-computers) provide multitasking capability which allows a number of separately compiled program modules to operate in parallel [VAX documentation 1984, Hoppe 1991, Olaleye 1992]. And since many GISs also include most of the editing capabilities needed for map editing and databanking, the system designer or analyst can exploit these to circumvent the isolated operation problem. For example, the CARIS server module has been developed by Universal Systems Limited (USL) to enhance the capability of interfacing the CARIS GIS to external software packages [Reeler 1990]. This facility enables users to interactively interface other program modules in order to have access to the CARIS data files for real time graphics display, editing, storing and retrieving of data. It provides software developers with many powerful capabilities which include:

1. storage and retrieval of CARIS GIS data directly using

simple subroutine calls.
2. access to all of the graphics functions of both CARED and CARMAN modules, allowing users to manipulate and display their graphics
3. access to database management systems, such as INGRES
4. access to continuous database functions
5. ability to incorporate topology into user applications
6. ability to customize the GIS application

In the system at U.N.B., this module provides a full programming environment and has formed the central tool in the solution of the isolated operation problem. Figure 2 shows a typical configuration including the server module.

Furthermore, both systems create a number of subprocesses which either run serially or in parallel with their parent processes, and which in turn may initiate subprocesses of their own, thus imposing an additional burden on the computing device. Employing this mechanism, the CARIS GIS invokes the host program as a child (the digitizing program, which also initiates its own subprocesses) and links it through the server module in order to share data in real time. Therefore, given the limitations on the number of concurrent processes that may be initiated from a single program, the host program has to be designed so that it does not initiate too many of them in order keep within the allowed limit. This implies that less modularization of the host program is necessary. On the other hand, because of the limitation on the stack memory made available to a single program, modularization has to be done to get the system working. This also simplifies the program development and maintenance. However, with the conflicting limitations, the challenge in the development of the system lies in keeping within the limitations of the computing resources without compromising the flexibility of the system.

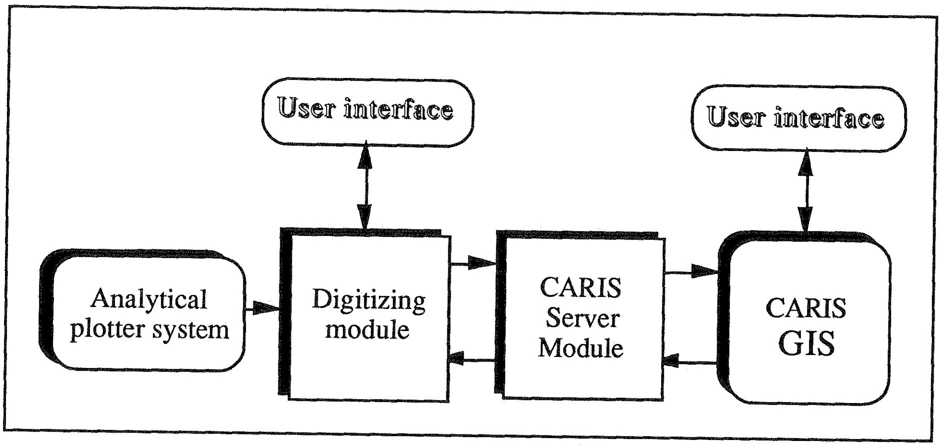

Figure 2: A configuration for the integration of the DSR-11 into the CARIS GIS

2.3 Limitation of the host computing facility

Although, the power of the modern computing facilities is increasing, certain tasks exert so much demand on such resources and thus exhaust the capability of the host computing device, particularly on single processor mini- and micro-computer based systems. Within the VAX VMS operating system environment, a program may be invoked from another program and it is referred to as a subprocess (a child) to the program which invokes it (the parent). Furthermore, an invoked program may invoke another program, which in turn may invoke another program and so on. The number of levels of such programs that may be cascaded is determined by the type of authorization and privileges given to the user. The CARIS GIS and the modules, which control the basic operations and applications of the analytical plotter system both demand high computing power.

To achieve this objective, the digitizing module of the UNB system was partitioned into **parent** and **child** submodules linked together by a buffer unit (the mailbox). While the parent cooperates with the CARIS GIS, the child and its associated subprocesses interact with the analytical plotter system. This approach allows the child digitizing module (CDM) to perform a variety of functions as necessary for operational flexibility while its parent (PDM) communicates and operates in parallel with the CARIS GIS. The CDM contains a large number of

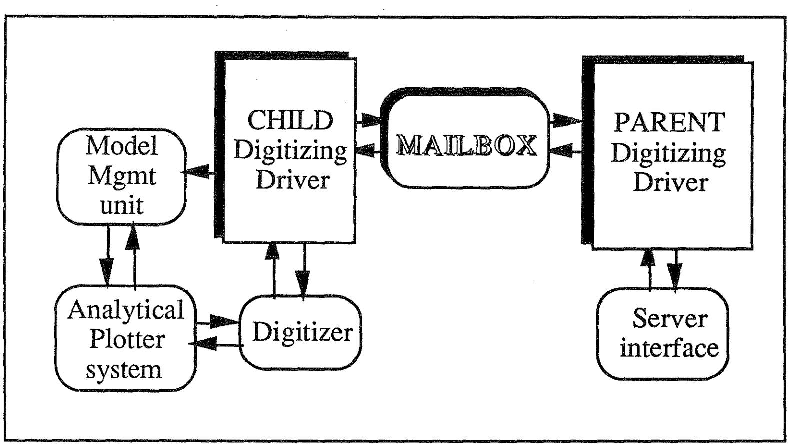

Figure 3: Partitioned configuration of the digitizing unit

the basic photogrammetric functions and so it is the active partition. The PDM is the dummy partition since it does not contain the actual codes for the digitizing tasks but small code segments which merely communicate with the server module and the child partition. In this way, the effect of the analytical plotter operations and its applications is hidden from the GIS operations while they share data in real time. Figure 3 shows the partitioned configuration, and Figure 4 shows the resulting configuration as implemented.

then wait for the cooperating image to perform the opposite operation. **Immediate** mode enables a program to perform a read or write operation to the mailbox and then continue to execute after the read/write operation is completed. **Asynchronous** mode enables a program to queue a read or write request to a mailbox and continue program execution while the request is being processed; and when the request is completed, a signal is given and the requested information is available to the process. Synchronous mode is nevertheless the easiest and often the most suitable method of addressing a mailbox when real time communication between two processes is intended. In a digital data collection and editing situation which involves different processes, real time data and command transfer is a requirement; and for efficient operation, such communication has to be synchronized; therefore synchronous I/O is the appropriate choice when one program reads or writes information to another image and cannot continue until that image responds.

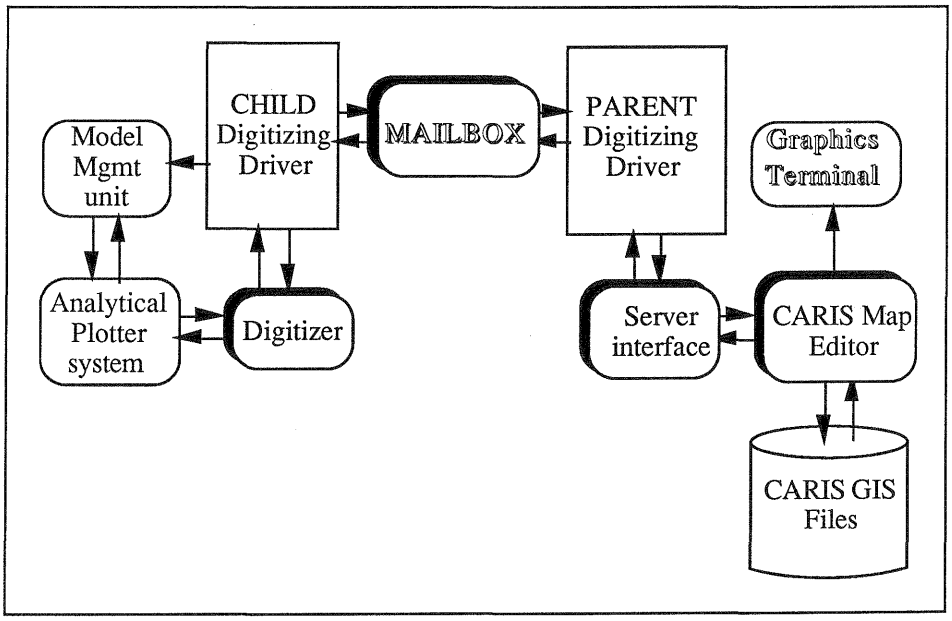

Figure 4: Implemented system configuration.

In practice, the mailbox is operated like any other file in the program. Once it is opened, FORTRAN formatted sequential read and write statements may be used for the I/O operations. The VAX system automatically synchronizes the I/O by not allowing a program to complete an I/O operation until a cooperating program has performed the opposite operation. For example, if one program performs a mailbox read operation, control is not returned to that program until a cooperating program performs a write operation to the same mailbox.

3 PASSIVE SYSTEM INTERFACES USING THE MAILBOX CONCEPT

To facilitate the real time communication involved in the system, the VAX operating system provides a number of communication options (passive interfaces). In general, symbols, logical names, mailboxes, installed common blocks, and global sections allow the passage of information between programs executing in different processes. In particular, logical names are used to pass brief messages from one program to another. Mailboxes, installed common blocks, and global sections are used to carry on a dialog between programs. The longer the messages in the dialog, the more reasonable it is to use installed common blocks or global sections. However, the communication requirements between the parent and the child digitizing processes are such that a record of information at a time is all that is passed, so the mailbox option is appropriate in the circumstances of the system discussed here.

The word "mailbox" brings to mind the idea of a post office facility through which mail is received. In the context of computer data processing however, it is a buffer unit which contains a single record into which information is written by one program for another program to pick up. In practice, a mailbox is created through a call to the VAX system service routine from within a program. The operating system creates the requested mailbox and returns the number of the I/O channel assigned to the mailbox and also its logical name. All the cooperating programs may then open the mailbox for read and write operations.

In general, there are three ways by which the mailbox may be read from or written into. **Synchronous** mode enables one program to perform a read or write operation to a mailbox and

4 THE USER-INTERFACES

User interfaces are an essential component of any integrated user-oriented system. Being the medium of communication between the user and the computer, its impact on the efficiency of operation of the entire system is tremendous. The human factor is still an important requirement in today's spatial data acquisition, management and application systems; the primary role being to supply the high level intelligence needed in certain critical operations and to direct and control the system towards achieving the intended goal. Obviously, the productivity of the operator and thereby the extent to which system objectives are realized are dependent on the ease and flexibility provided by the user interface.

The process of acquiring digital data from an analytical plotter involves a number of complex and interrelated operations. Using an integrated system, the operator has to initiate the execution of many programs either for in-line operation or for concurrent operation, and each program presents him with its own interface of optional menu items from which a choice must be made in order to achieve the desired result. The way in which these various menus are organized into logically related classes to facilitate easy use by the operator is extremely important. Furthermore, the menus at various levels must be synchronized into a logical piece so that it is not apparent to the user from which module the menu is being displayed.

There are four levels of system-user interactions involved in the integrated system that is being discussed (Figure 5). These include (1) the interface of the CARIS GIS (2) the parent digitizing interface (3) the child digitizing interface and (4) the photogrammetric workstation interface.

contains all relevant CARIS GIS line digitizing commands would be displayed at the child driver menu. Practically, the purpose of this menu level is to coordinate the activities of the various subroutines and subprocesses which actually execute the designated tasks. This interface, in essence, brings the user close to the functional level where tasks and options are more specifically defined. Indeed, the selection of a menu option at this level has the effect of activating the particular subroutine or subprocess that is designed to handle the option selected.

Fourth level user interface: This is a collection of user-interfaces offered by the modules (MACROS) which operate at the photogrammetric workstation. These include those offered by the actual digitizing subprograms and those of the photogrammetric sub-processes.Usually, these interfaces, being at the operational level, provide options which may be used to specify how the task is to be performed or to override certain actions of the operating program, or to change the type of data to be collected etc. For example, when the point-to-point digitizing MACRO is activated, the options at the operational level will include (1) deletion of last digitized point, (2) digitizing a 3-point arc, (3) quit this command etc. Other tasks have their own menu options as well. In general, when an option is selected, it is immediately sent to the CARIS editor through the mailbox and the server interface. The CARIS GIS task module then performs the specified operation immediately. On the other hand, the purely photogrammetric menus often lead to a number of other lower-level interfaces. offering menu options related to the photogrammetric tasks. In all situations however, the user is returned to the CDM driver interface after each task is concluded.

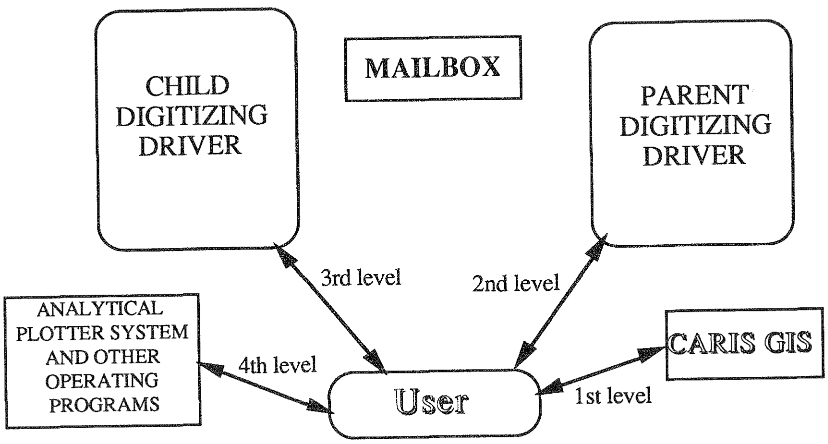

Figure 5: Computer-user interfaces for the system

First level user interface: This is offered by the CARIS GIS in which the user talks to the GIS directly. All the GIS applications are directly available at this interface in which the user communicates with the program through menu selections.

Second level user interface: This is offered by the parent digitizing driver module (Fig. 5). In this interface, communication is achieved by menu selection. Included in the menus are options such as LINES for the digitization of line features, NAMES for the addition or deletion of feature names, SYMBOLS for the manipulation of symbols, and options relating to the use of the analytical plotter system for the restoration and interchange of stereomodels etc. When the user selects an option, the selection is immediately sent to the child digitizing unit through the mailbox, from where other lower-level menus are displayed. In some applications, the selected option is completely processed by the parent driver module. For example, when the user selects the option for changing the scale of the graphics display, the CARIS GIS command "MAd" is immediately sent to the CARIS editor via the server interface and then both the old scale and the prompt for the new scale are displayed. Upon the user entering the new scale specification, it is immediately sent to the CARIS editor to effect the desired change. In general, most of the options at this level are meant to initiate the child process to perform a specified task.

Third level user interface: This is offered by the child digitizing driver module. The menu items at this level are more specific than at the parent's level. The menus are arranged in groups, and the particular group of items displayed at any time depends on the option selected by the user at the parent's menu level (Figure 5). For example, if the user selected option LINES from the parent's menu, then the menu group which

5 OPERATIONAL COMMUNICATION MECHANISM OF THE SYSTEM

To demonstrate the operational communication mechanism of the integrated system developed at U.N.B., two example uses are given. The first example shows the communication mechanism between these two units when a user requests to digitize a line in a point-to-point mode by issuing the appropriate CARIS GIS line command. At the parent digitizing process end, the user selects the menu option corresponding to the line digitizing command; the parent process then writes this command to the mailbox and waits for a response from the child process before continuing with the execution. The child process reads this command and then calls the appropriate subroutine that initializes the digitizing process. The digitizing subroutine displays a message to the user requesting the user to specify the feature code and other parameters of the graphic element to be digitized in the order in which such information is normally requested in the CARIS GIS environment. This information is then written to the mailbox, and control returns to the parent process which then reads the mailbox and transfers the information to the CARIS GIS through the server interface. Control is then returned to the child process in which the line digitizing routine (C_LIAP) subroutine initiates the collection of the stream of X,Y, (Z) coordinates from the stereo digitizer. These data are then written to the mailbox and read by the parent program, one record at a time until the operator terminates the process. Normally, when the user wants to change a process or terminate the execution of a command or delete a digitized point, this is indicated by pressing the footpedal of the analytical plotter, after which a menu is displayed giving the user a number of possible options for the particular command currently in progress. If the user intends to terminate the command for example, the **Quit** option is selected and the child process sends the appropriate message to the mailbox, and control returns to the parent process which then reads this information from the mailbox. Other commands may be similarly selected and processed. One unique feature of this process is that as the messages are transferred from the child to the parent process, each message is immediately analyzed by the CARIS GIS, and the appropriate action is taken such as drawing the element on the graphics screen in real time or perform other GIS functions.

The second example demonstrates the situation between the child and the parent digitizing processes when the operator chooses to change to another stereomodel. From the parent process interface the operator selects the menu option corresponding to the "change model" command. This command is then written to the mailbox for the child process. Reading the change model command, the child process initiates the procedure for model restoration by calling the model management and restoration program into action. The model restoration process involves a complex network of interrelated tasks which may include re-initiating the analytical plotter, allowing the operator to perform a whole range of operations such plate registration, image space mensuration, relative and absolute orientations, each of which is controlled by separate dedicated program modules. Therefore, the model management module has to invoke other processes to carry out these functions at the request of the operator. In the meantime, since nothing can be done until a new model is restored, the child process has to wait for these other subprocesses to conclude the model restoration task and return a message to it before it can continue. Therefore, since the child process and these other subprocesses operate sequentially, real time synchronous communication is not required between them. In any case, after the model is successfully changed, the child process writes a job-completed message to the mailbox and control returns to the parent process for subsequent operations.

These two examples demonstrate the communication mechanisms that exists between the photogrammetric digitizing system and the CARIS GIS using process partitioning and the concept of mailboxes. Central to the successful use of the system are the various user interfaces and menus, through which the operator communicates with the system to control the operation. The logical consistency and user friendliness of these interfaces are vital to the productivity of the operator.

6 CLOSING REMARKS

Direct integration of the photogrammetric workstation to a GIS has many advantages which will increase with improvement in digital technology. However, the problems of data structure incompatibility, "closed" operation of the GIS package and limitations of the available computer facilities militate against the full realization of such integrated systems particularly when parallel operation is desired. Programming interfaces such as the CARIS server module, which are now provided by most GISs and the mutitasking facilities now available in most operating systems, provide a powerful means for solving these problems.

REFERENCES

Hodgson, M.E., Barrett, A.L., and R.W. Plews (1989). "Cartographic Data Capture Using CAD", Auto-Carto 9 Proceedings, Ninth International Symposium on Computer-Assisted Cartography, Baltimore, Maryland, pp. 357-366.

Hoppe, A., 1991. "Concurrent and Distributed Programming in UNIX Environment", International Journal of Mini and Microcomputers, Vol. 13, No. 3, pp. 109-116.

Olaleye, J., 1992. "An Investigation into the Optimum Software Architecture for an Analytical Photogrammetric Workstation and its Integration into a Spatial Information Environment", Ph.D. Dissertation, Department of Surveying Engineering, U.N.B.

Ramirez, J.R., 1989. "Understanding Universal Exchange Formats", Technical Papers, ASPRS/ACSM Annual Convention, Baltimore, Vol. 2, pp. 108-116.

Reeler, T., 1990. "CARIS Server Interface", Universal Systems Ltd., Document No. SERVER-JAN90-VMS, 14p.

VAX Documentation, 1984. "Fortran Programmer's Manual", No. AA-Z212A-TE, Unpublished.