IBM CONFIDENTIAL

L. E. J.

MAR 28 1973
from Mike Kutcher

UC

IBM CONFIDENTIAL

COMMON

SDD/GSD INTELLIGENT SUBSYSTEM SUPPORT

FUNCTIONAL REQUIREMENTS

REVISION 1

R. A. Hase

IBM CONFIDENTIAL

IBM CONFIDENTIAL

THE JOINT SDD/GSD INTELLIGENT SUBSYSTEM SUPPORT FUNCTIONAL REQUIREMENTS STATEMENT WAS PREPARED BY B. MAYO, R. WINGO, J. WEBER, AND J. MARKS, WITH MUCH ASSISTANCE AND HELPFUL COMMENT FROM MANY OTHERS, AS AN ACTIVITY OF THE ISSS FUNCTIONAL REQUIREMENTS TASK FORCE, JANUARY 1973. IT IS A CROSS-DIVISION, CROSS-PRODUCT REQUIREMENTS STATEMENT AND IS INTENDED TO PROVIDE THE BASE FOR ESTABLISHING OBJECTIVES AND SPECIFICATIONS FOR IBM SUPPORT OF INTELLIGENT SUBSYSTEMS.

R. A. HASE

C. R. WHITE -

TABLE OF CONTENTS

IBM CONFIDENTIAL

TABLE OF CONTENTS (continued)

1.0 OVERVIEW

1.1 PURPOSE/SCOPE OF DOCUMENT

It is the intention of this document to present requirements of Intelligent Subsystem Support needed by the UC Family, Sensor Based products, small commercial/scientific, and S/370 subsystems.

Further, the functional requirements are presented with the requirement that the customer sees a level of support consistent with SNA and user oriented facilities such as high level language, CICS, IMS, et cetera that addresses the complete product line of subsystems.

SNA: Standard Network Architecture

The long range revenue of the corporation is heavily dependent upon broad expansion of the computer application base. With the continued technology studies yielding ever improving price/performance ratios, more and new uses for our products must be found. It appears certain that a key contributor to that expansion will be the continued advancement of terminal and sensor based system technologies. Making our systems more accessible and easier to use, to more people is fundamental to our growth; and is central to the requirements defined in this document. It is clearly our intent to establish a systems capability consistent with our current and planned product requirements, and to establish a systems technology base consistent with the known FS requirements.

Although sensor based products and UC based products each have some unique requirements, a task force study (Appendix 6.2) has established that both UC and Sensor Based products have common ISS requirements. These requirements are outlined in Section 2.0 and Section 3.0 of this document.

1.2 BACKGROUND

1.2.1 SENSOR BASE HOST SUPPORT EVALUATION

The support of Intelligent Subsystems is not new in IBM. There has been considerable experience, primarily in internal manufacturing, with the first systems implementation dating back to early 1965; the COMAT System (Computer Operated Manufacturing and Test System) developed by San Jose

1.2.1 (continued)

manufacturing. During about the same time frame, the CIMPAC system (Computer Integrated Manufacturing Process and Control) was being developed by Endicott manufacturing.

Those development activities led to the 1968 development of the common SMD/CD/WTC Manufacturing Process Control System, which included the development of an OS/360 subsystem called PC/OS (Process Control/ Operating System). That system is currently installed in over 20 internal locations world wide.

Other internal manufacturing developed ISS support systems include: DIMS, developed by San Jose Manufacturing and SIFEX, developed by Sindlefingen Manufacturing.

DPS
Distributed Systems Program

Product support experiences include 1130/DSP and 1800/DSP (Distributed Systems Program) which provided the initial System/7 host support, and 370/DSP which is planned for release in June 1973.

Since the introduction of sensor based products, the need for high level host support has been apparent as evidenced by the 1800 requirement of high speed attachments. Although the 1800 was not sold as a host attach device, customer demand forced the introduction of a TP link. At the present time, over 10% of the installed 1800s have a TP host link. In addition, devices such as the Sensor Based Control Unit (SBCU) was announced as an RPQ device to satisfy customer requirements of a high speed in-plant communication system.

The S/7 was announced with a marketing thrust of Host Attach, with over 50% of the orders having a TP link. In addition, many RPQ communication attachments are available to satisfy customer demand. A Distributed System Program (DSP) was announced to satisfy the ISS requirements of the sensor based products. This will provide the user with a high level get-put type language. DSP addresses not only program movement but data movement as well. DSP will continue to provide interim support for the sensor based products.

1.2.2 TP EVOLUTION

Start

The concept of a Universal Controller (UC) family to support and accommodate a variety of different devices is relatively new. The requirement for a terminal controller that can accommodate many different devices originated when it became apparent that hard wired control units, dedicated to one or a small number of terminals, contributed a burdening cost to the terminals it supported. The UC supporting an unlimited number of different terminals offered a way for terminals to share the cost of the controller thereby reducing the cost to each terminal. The early view of the UC, then, was a programmed equivalence of a hard wired controller.

The notion of intelligence, available with a UC, had not been fully considered or exploited within SDD until the ROCKET family of controllers and its terminals began definition. It became clear that intelligence was required for remote key entry, graphics design drafting, data collection, et cetera.

* * *

In 1968, the attachment of an 1130 with a 2250 IV graphic device to a S/360 host was shipped. The 1130 was an intelligent subsystem supported by callable subroutines in the 1130 to access the graphic functions and control data transmission to the S/360. S/360 support consisted of callable subroutines to communicate with the subsystem. The product missed forecast. The level of programming support was a key contribution to the failure.

The access method support for intelligent controllers and their terminals required standardization to facilitate host to subsystem and subsystem to terminal communications. Support of UCs through a variety of access methods and line control disciplines was not viable. Standard Network Architecture which encompasses VTAM and SDLC was defined to assume a consistent viable level of communications among Host, Subsystem, and Terminals.

SNA
Sdd Data Link Control
(next pg)

Sensor Base and Internal Manufacturing use of intelligent control units offers a level of experience acquired during a seven-year period that

1.2.2 (continued)

is the basis for the requirements described in this document. The application environments where UCs will be installed are identical or similar in many cases to those in which S/7s exist. Now, the announcement of Virtual Systems as well as the strategy to migrate all customers under the standard network architecture presents an opportunity to provide consistent and complete Host support for all IBM intelligent subsystems.

1.3 RELATED PRODUCT ACTIVITIES

There are several current product definition and development activities that directly relate to the Intelligent Subsystem Support requirements outlined in this document. They are listed here for information purposes, with no intent to imply any specific implementation approaches.

DB/DC

(A) DB/DC, in particular IMS and CICS. In addition to the Data Base and Data Communication functions the current plan for Release 3 of both products will include specific support of System/7 as an "Intelligent Subsystem". RTS-3 will also be supported by IMS/CICS as an intelligent subsystem in the 10/74 time frame.

370/DSP

(B) 370/DSP (Distributed Systems Program). This subsystem is currently planned for initial release in June 1973 on OS/MFT using S/S lines as a Type-1 extension. It is intended for interim support of remotely attached System 7s, and will provide a very comprehensive level of "Intelligent Subsystem" support functionally equivalent to the ISS requirements stated here.

SNA

(C) SNA (Standard Network Architecture). SNA encompasses a total communications strategy which will heavily influence (or be influenced by) the ISS requirements.

SDLC

(D) SDLC (Standard Data Link Control). SDLC is the single TP line discipline chosen for SNA.

VTAM

(E) VTAM (Virtual Telecommunications Access Method). VTAM is intended to be the sole supported TP access method in the future (S/370), and will be the primary implementation vehicle for SNA support.

SSS

SIA

VSAM

FS

1.3 (continued)

(F) SSS (Subsystem Support). This is the Intelligent Subsystem support base currently being developed for UC products. Industry and cross industry unique code will interface with this base.

(G) SIA (Systems Interchange Architecture). Defined an inter-system communication discipline. 370/DSP supports a subset of this architecture.

(H) VSAM (Virtual Storage Access Method). VSAM is intended to be the base for all future direct access storage support. Future releases of IMS, CICS, and DSP will migrate to VSAM as soon as practical.

(I) FS (Future Systems). Efforts are currently underway in developing a Sensor Based/Intelligent Subsystem support strategy. (Reference 6)

1.4 ENVIRONMENT

Sensor Base only

The marketing environment for Intelligent Subsystems has been to sell a top down approach. That is, implement the data base on the host and allow the remote users access to that data base. Not only allow use of the data base but also collect information from that subsystem to update the host data base. Without the provided ISS support, the customer or industry group must write their own SCP-like support. Not only is this time consuming, it requires a high degree of skill. The result is the customer's growth is stifled, which means IBM's growth is stifled.

In this complex environment, we find customers needing the power of 370s with attached Intelligent Subsystems to perform such tasks as inquiry, data entry, editing, and controlling.

Although requiring a good response (1 to 2 seconds) for the first three application areas, the fourth area, control, may dictate a higher response requirement. For example, a production monitoring job controls an entire plant process, dictating decisions that must be made by the host and transmitted to the subsystem in one second or less response. If not, the entire production line may stall.

1.4 (continued)

The indication given the subsystem by a plant floor device may require transmission of pertinent data to the host, calling and executing a program in the host and passing the desired results back to the subsystem. This transmission could also mean passing of new data as well as a program to be executed in the subsystem, in order to keep the production line running.

Management needs "real time" reporting capabilities as to how the process is performing so that decisions can be made that will favorably affect the output of the product. The need for timely information is, in many cases, the prime justification for a hierarchical system. For example, twenty-four hour old information is of little value in a plant floor environment.

In order to implement in this type of environment two items are needed: An intelligent subsystem and the intelligent subsystem support that will allow customers to implement easily and within a reasonable time frame.

1.5 REFERENCE DOCUMENTS

1. Subsystem Support Services for Intelligent Terminals, Preliminary Final Programming Functional Specifications; November 22, 1972; Document No. OS-3013-00-RAL

2. S/370 Distributed System Program/OS, Functional Specification Manual; October 18, 1971

3. Programming Objectives IBM S/370-DOS, AOS/1, AOS/2; October 8, 1971

4. Final Programming Functional Specifications, IBM S/370-AOS/1-Integrated Sensor Base Support; S/370-OS-2061-01 END; December 22, 1971

5. Standard Network Architecture: System Description SNA-0010-00; January 4, 1973

6. FS Sensor Based Objectives and Strategy, Preliminary; R. R. Guyette; January 1973

7. SRI Study - The Use of Hierarchy Computer Systems - Final Oral Presentation, December 12, 1972

8. SRI Study - The Growing Market for Minicomputers; March 19, 1971

9. 1970 DP Group Net Potentials Study

10. 1972 WTC Long Range Market Potential Study

11. IBM Journal of Research and Development, Volume 14, Number 6, November 1970; Set of Articles on Internal IBM Manufacturing Distributed System Development

12. Memo to File; May 18, 1972; J. W. Marks; "Trends in Instrumentation and Hierarchical Systems"

13. Memo to File; June 23, 1972; J. W. Marks; "Manufacturing and Process Industries Sensor Based Market"

14. EDP Industry Report; November 30, 1972; Volume 8, Number 2

15. SRI Study - Analysis Selected USA Instrumentation/ Minicomputer Market Segments; May 1972.

1.5 (continued)

16. Preliminary Sensor Based Market Requirements, September 30, 1971. D. Ashley, R. Highwart, D. Loposer, H. Rojas

17. Memo to Distribution. Minutes of 12/19/72 Sensor Base Coordinating Council Meeting, R. Hase

18. Final Program Specifications: VTAM, S/370-OS-2086-04-KGN, November 27, 1972.

19. Rocket Forecast Assumptions, Phase 1, KGD-0194-00-676; J. J. McCormick

20. Initial Program Specifications: GCP, CPS-1535-02-RAL; September 15, 1972

2.0 GENERAL INTELLIGENT SUBSYSTEM SUPPORT REQUIREMENTS

2.1 HOST SUBSYSTEM RELATIONSHIPS

To understand the ISS support requirements in a distributed system environment, it is useful to assess the various ISS roles in the overall system. (See Figure 2A) As shown, an ISS may be used to satisfy a number of different functions, any of which may be "system controlled" or "user controlled"or both. Complete ISS support must satisfy the needs of all these various situations, as follows:

System versus User Oriented

The use of an ISS in system oriented roles, such as a system I/O controller, line concentrator, or fixed function terminal, presents much the same support requirement as a non-ISS based unit, except for the initialization functions (program preparation and load).

Support of an ISS in a user oriented role (with the user defining its functions and writing application programs), requires considerably more comprehensive support; addressing such considerations as linking distributed application modules, and providing user data services.

IBM versus User Programmed

The IBM programmed ISS can be viewed essentially as a fixed function unit from a system support point of view (again, except for the initialization functions). The user programmed ISS must be supported in much the same manner as if the application were being implemented in the mainframe.

Loosely Coupled versus Tightly Coupled

In a loosely coupled environment, the ISS is essentially independent of the host with the more frequently used I/O devices locally attached, and generally communicating with the host on an infrequent basis. In a tightly coupled environment, the ISS is dependent on the host for most I/O services (files, printers, etc.), for compute capability (Multiply/divide, floating point, etc.), and may even be an extension of the host application. High speed communication requirements are most often associated with this environment, such that ISS to host access approaches local direct access storage performance. In actuality high speed requirements exist independent of degree of coupling and vice versa. High performance loose coupled hierarchical structures are typical of the in-plant environment.

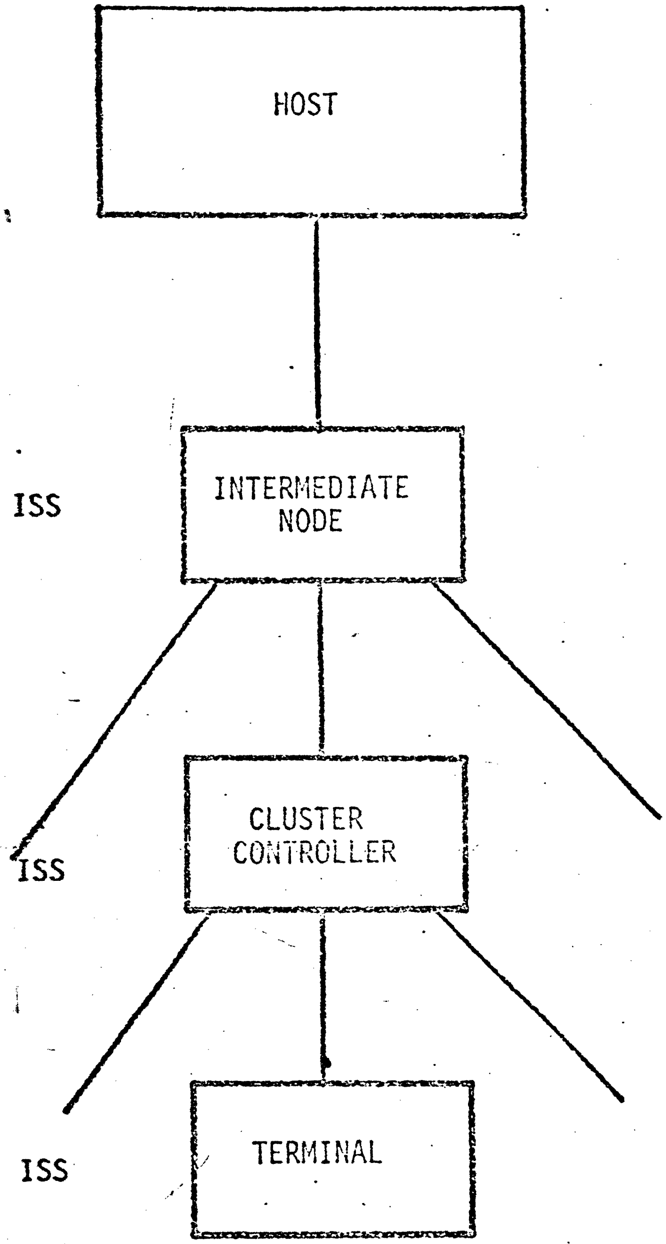

FIGURE 2A

ISS ROLES/SNA

2.1 (continued)

Attended versus Unattended

Most user oriented ISS units operate in an attended mode. Initialization, set-up, operation, and exception-handling are operator functions; and system support must be oriented to interfacing with an operator controlled device. Many system oriented ISS units operate in an unattended mode. All initialization, control, etc. must be provided by the host with no host operator intervention other than that associated with a similar non-intelligent controller. Some user oriented ISS units also operate in an unattended mode; such as a sensor-based monitoring system or a Rocket 3 located at a remote/un-manned site. For unattended mode support, function must be provided to allow complete host application control of the remote ISS.

TP versus In-Plant

***

TP communications must, of course, be used to link remote/off site ISSs with their hosts. TP will also be the selected media where ISSs must be linked together in a "network" fashion, or where performance requirements are not demanding and cost trade-offs favor TP over an in-plant media. TP linked ISSs will normally be loosely coupled due to the relatively slow transmission rates. In-plant high-performance communications will be favored in most installations where geography permits.

The high performance selection will be made because of:

(a) The need to implement tightly coupled host control

(b) The need/desire for a responsive/high-performance installation. (unknown future requirements for response)

(c) Large data volumes.

(d) Aggregate heavy traffic/data volumes due to a large number of host connected ISSs.

(e) The need to minimize ISS cost by centralizing common functions, such as DP I/O. (cost trade-off or enivronmental constraints on DPIO)

(f) Communication cost advantages (large number of ISSs or mix of TP and high speed requirements where using a single type minimizes cost)

2.1 (continued)

The current Standard Network Architecture definition is comprehensive in addressing the various TP connect functions, but must be expanded to properly address the in-plant high-performance requirement (the SBCU, 3272, et cetera).

It should be noted that the in-plant high-performance requirement is not limited to only sensor-based system communication. It applies to non-sensor based systems as well, i. e. large installations, graphics, and mixed sensor-based/non sensor-based environments (this is the normal system environment--all sensor-based applications have man-machine interface requirements as well as machine--machine, though some of those requirements are best satisfied with locally attached terminals).

2.2 USER INTERFACES

The user interface that immediately comes to mind is that associated with data interchange between the host and the intelligent subsystem. Satisfying only that interface requirement, however, does not allow for effective implementation of distributed applications (interacting user function in both the host and the subsystem). Three interfaces must be provided for support of ISSs (see Figure 2B):

Interface 1

This interface (1a) provides for host-ISS interaction, and includes functions such as Program Load, Program Start, Data Get/Put, etc. It also allows (1b) for direct communication between host application programs and subsystem application programs.

Interface 2

This interface allows interaction between a single user written program and the standard ISS support functions without requiring an application in the other system/subsystem; such as a user program at the host reading a data set written by the subsystem (which may be physically resident in either a host file or a subsystem file).

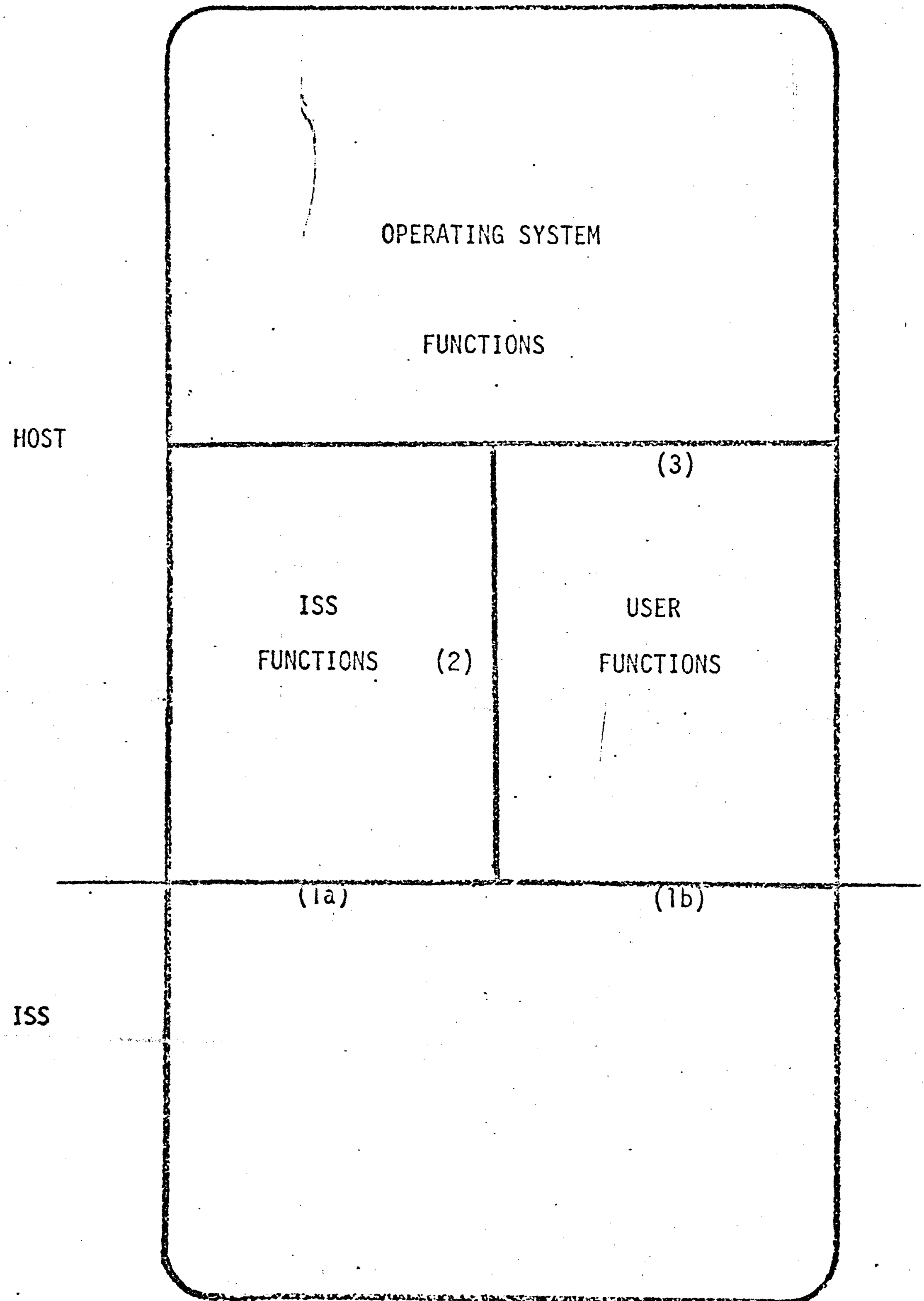

FIGURE 2B

USER INTERFACES

2.2 (continued)

Interface 3

This interface provides for interaction between the foreground user task/programs and the rest of the system; (For example passing data to another region or starting a background job.) Implicit is the need to provide for multitasking with performance consistent with the communication media and real time responsiveness. The real time ISSS user must have full access to all services provided by the operating system function.

2.3 The functional support requirements for Intelligent Subsystems are summarized in the following table. Each of these is addressed in detail in Section 3.

2.3.1 SUBSYSTEM PROGRAM LOAD

(A) Initial (Bootstrap)

(1) At Request of:

(a) Host Application

(b) Target Subsystem Application

(c) Different Subsystem Application

(d) Target Subsystem Power-On or IPL Button

(2) Source of Program Load is:

(a) Host Library

(b) Target Subsystem Library

(3) Controlled Termination

(B) Program Load (Overlay)

(1) At Request of:

(a) Host Application

(b) Target Subsystem Application

(c) Different Subsystem Application

(2) Source of Program Load is:

(a) Host Library

(b) Target Subsystem Library

(3) With or Without Execution

(4) With or Without Associated Data Transmission

2.3.2 INVOKE/SYNCHRONIZE PROGRAM EXECUTION

(A) In Host from Subsystem Application

(B) In Host from Host Application

(C) In Target Subsystem from Host Application

(D) In Target Subsystem from Another Subsystem Application

(E) Without Partition/Region Restraints

(1) Subtasks of Real Time Job

(2) Other "On Line" Partitions/Regions

(3) Background (Batch)

(F) With or Without Associated Data Transmission

(G) Transparent to Paging, Library References, etc.

(H) Program Controlled Event Posting

(I) Timed Interval/Event Basis

2.3.3 TIMER SUPPORT

(A) Invoke/Synchronize Host/Subsystem Application Program

(1) Starting at "n" O'Clock

(2) Every "n" Time Units

(3) Until Terminated by:

(a) "M" Repetitions

(b) User

(B) Using Parameters Initialized/Modified by Host/Subsystem Application Programs

(C) With Access to Time-of-Day Clock from Host/Subsystem Application Program

(D) Synchronized to External User Clock

(E) Watchdog Timer Services

2.3.4 DATA INTERCHANGE

(A) Between User, Industry, or SCP Programs

(1) In Host

(2) In Host and Subsystem

(B) Between Programs and Named Data Sets

(1) In Host

(2) In Host and Subsystem

(3) In Subsystem

(C) Between Host or Subsystem User or Industry Programs and Host Data Base

(D) From Subsystem Application Program to "System File" (card/printer I/O).

(1) Open and Close File (release to punch/print)

(E) With Option for Program Execution

2.3.5 DATA SET/PROGRAM LIBRARY MAINTENANCE

(A) Create/Delete/Maintain Named Data Sets

(1) On Host at request of Host Application

(2) On Host at request of Subsystem Application

(3) On Subsystem at request of Host Application

(4) On Subsystem at request of Subsystem Application

(5) Main storage and secondary storage resident files

2.3.5 (continued)

(B) Create/Delete/Maintain Subsystem Program Library

(1) On Host at request of Host Application

(2) On Host at request of Subsystem Application

(3) On Subsystem at request of Host Application

(4) On Subsystem at request of Subsystem Application

2.3.6 CONNECTION MODES/SPEEDS

(A) Support full range

(1) Common Carrier

(2) In-plant

(B) Allow Integrated and Standalone Control Units

(C) Provide System Responsiveness commensurate with line speed

(D) Mode/speed transparency required

2.3.7 OPERATIONAL REQUIREMENTS

OLT ?

(A) 24 hour, 7 day Environment; RAS, OLT, Recovery Requirements

(B) Unattended Subsystem Operation

(C) High Volume, multiple subsystems on host

(D) High Performance target:

Host response equal or faster than subsystem disk

2.3.7 (continued)

(E) User interface consistent with High Level Languages

(F) Dynamic Library and Data file replacement

(G) Security: Unauthorized Access Control

3.0 DETAILED REQUIREMENTS

The wide range of application environments and planned intelligent subsystems creates the need for several options in each of the subsystem requirements listed in Section 2.3. This section will define those options, discuss the application characteristics, and point out known differences between sensor base and terminal oriented subsystem requirements. A tabular comparison is provided in Appendix 6.2.

3.1 SUBSYSTEM PROGRAM LOAD

(A) Initial Program Load

Many subsystems will operate in an environment tightly controlled by a host application program. Typically each subsystem will be IPL'd as a result of the initiation and/or execution of the host application. To meet the variant needs of a range of subsystems on option is necessary to specify whether the subsystem is to IPL from its own local secondary storage or receive the IPL via the connecting data link. These requirements are common to both UC and S7 planned products.

In environments (typical of Sensor Base) where the subsystem is running an essentially standalone application with only a loose-coupled realtionship to a host application the need exists for the subsystem application to determine the need for IPL. This decision could result from error recovery procedures or from recognition that a new application requiring a new SCP should be loaded and executed. In either case the function required is initiation of an IPL sequence from application code running in the subsystem. An option is required to specify the source of the IPL data.

SCP (System Control Program?)

In another environment multiple subsystems may be in use with one specific subsystem providing a master coordinator or controller role. This environment, found in both sensor base and terminal applications, requires the ability for an application running in one sybsystem to initiate an IPL sequence in one or more other subsystems. Again an option for the source of IPL data is required.

3.1 (continued)

The controlling subsystem should be able to direct each subsystem to IPL from its own secondary storage, or alternatively should be able to invoke IPL of the other subsystem(s) from a host library. Two possible implementations of this latter capability are:

(1) The controlling subsystem could command each of the other subsystems to IPL themselves, or

(2) The controlling subsystem could invoke the host to IPL the other subsystems.

Either would meet the requirement.

A third environment will require that uninitialized subsystems be IPL'd either as a result of operator pushbutton action or as a result of power on or error recovery conditions in an unattended subsystem. Unattended subsystem operation is particularly important in sensor based applications involving configurations distributed over large areas such as oil fields, pipelines, power distribution systems and even some large factories.

In applications typified by host or "master subsystem" control a probability exists that running subsystems will be re-IPL'd. In consideration of this, implementations of the IPL function should provide for a controlled termination of the subsystem. Since the exact state of the subsystem (error loop or normal execution, etc.) cannot be determined, the controlled termination may have to be limited to a "subsystem reset" and the user advised to plan his application accordingly.

(B) Program Load

The functional requirements for subsystem application program load parallel the IPL requirements. The program load or program overlay should be capable of initiation from the host, the target subsystem, or a different "controlling" subsystem. An option should be provided to load the program from the subsystem library or from the host library via the data link.

3.1 (continued)

The controlled termination aspect of the IPL requirement is not applicable to application program load or overlay. Options are required to specify immediate execution or "wait", and to provide at least a limited capability to append data to the program when it is loaded via the data link.

3.2 INVOKE/SYNCHRONIZE PROGRAM EXECUTION

Implementation of distributed system applications requires the capability to invoke or synchronize execution of a program in any system/subsystem from another system/subsystem. The "execute program" or "post task" function that the user sees should make paging, library references, etc. transparent. From the host the user should be able to specify execution of a named program or trigger continuation of a priority execution level/sublevel or branching into a code string. From the host or the subsystem, the user should be able to cause host execution of a named program as a subtask of the real time job; in another partition/region, or as a batch background job. He will also need to post waiting programs. Paging in of the program, retrieval from the library for transient execution, etc. should not require additional user intervention.

The function providing program invocation should optionally allow an associated data transmission. This option will let the user provide data required by the target program at the same time it is invoked. Because a corollary capability is provided in the data transmission function, the length of data appended to the execute program transaction may be restricted. In cases where significant amounts of data are required the data transmission function could be invoked with an optional request for program execution.

As an extension to the program invocation/synchronization function the SCP support should utilize the timer function addressed in 2.2.3 and 3.3 to trigger program execution based on user supplied parameters. This function should provide execution at specific time of day and at specific time intervals.

3.3 TIMER SUPPORT

The user requires a capability to cause predetermined applications to be executed at pre-established times. He can provide parameters (program name, subsystem number, time, etc.) describing the desired program execution. The intelligent subsystem support SCP should accept and use these parameters to invoke execution.

Additionally the SCP should provide user access to the time-of-day from host or subsystem user applications for time stamping, et cetera, and should also provide for synchronization with an external user clock. In many applications the subsystem activities need to be synchronized with human activities based on a centralized "plant clock".

In general it is expected that high resolution timing would be provided by individual timers in the subsystems. The timer support functional requirement is not applicable to these high resolution timers but to a "time-of-day" clock in the host.

A "watchdog" capability is required so the user can establish time-out limits to protect against program loops, etc.

3.4 DATA INTERCHANGE

User application programs, Industry Code, and SCP support running in either host or subsystem need to get data from and put data to named data sets and other programs regardless of whether they are resident in/on the host or the subsystem. To provide maximum usability of the functions provided by ISSS the interfaces shown in Figure 3A are are required. The user code will need to interface to ISSS for example to get or put data to/from a data set or program in the other system/subsystem. Industry code will have a similar requirement. It is probable that industry code will provide the user more complex function such as "get from subsystem data set and put to host system data set" as a single transaction. This would be accomplished by the industry code stringing or chaining together multiple requests to the ISS support.

It is likely that the ISS support will not exist as a single package but will be distributed through multiple

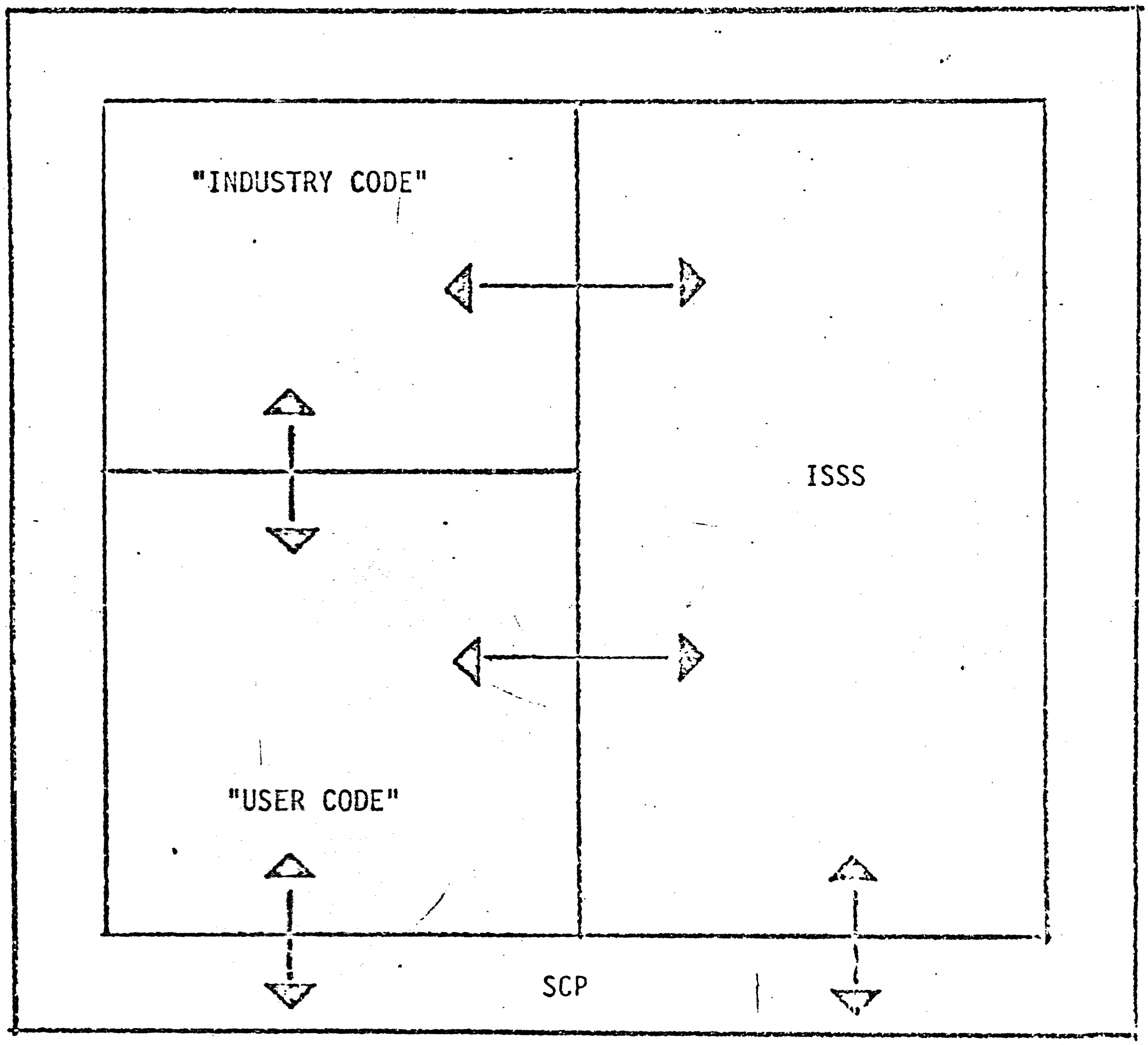

FIGURE 3A

INTERFACES

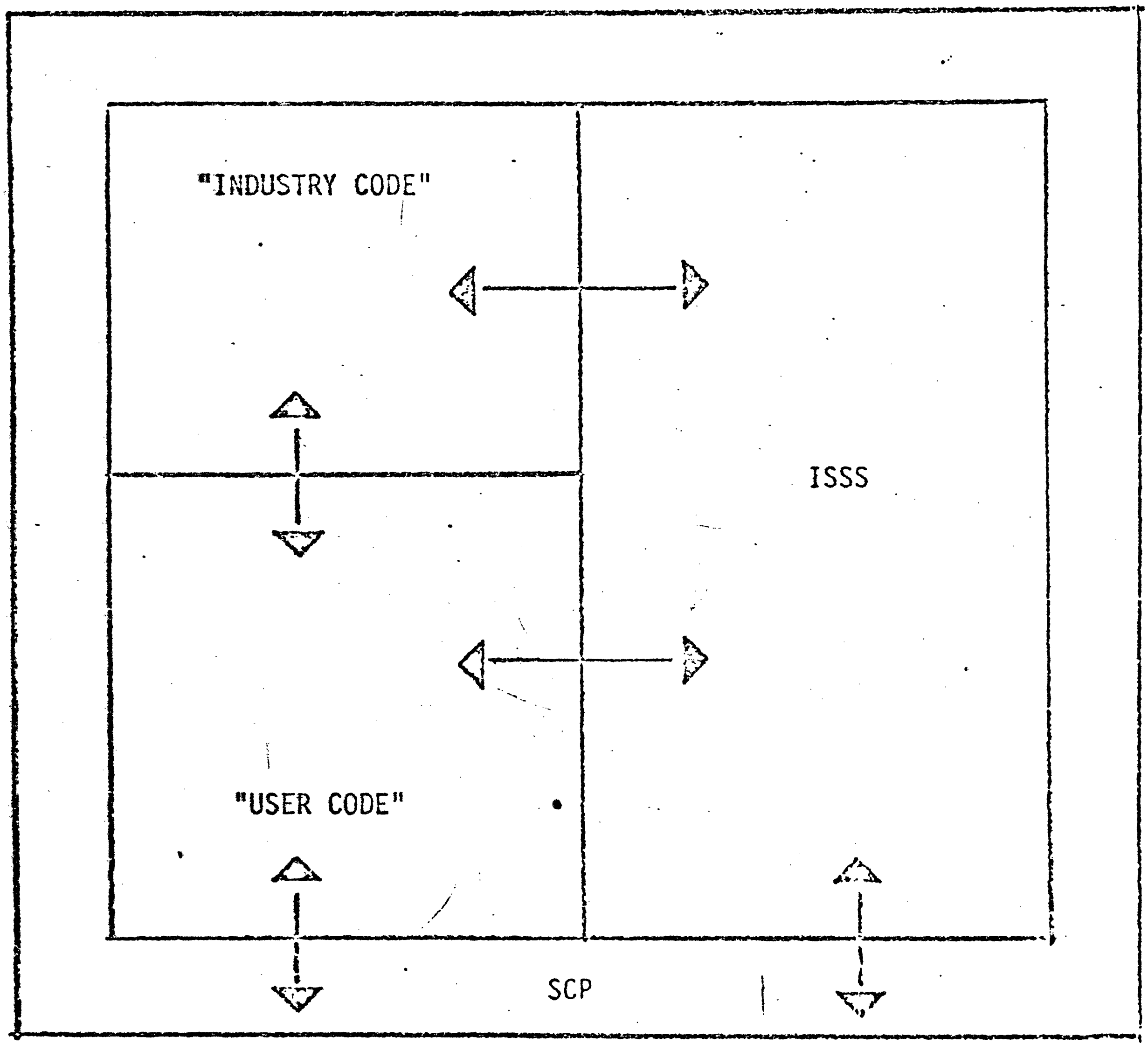

FIGURE 3A

INTERFACES

3.4 (continued)

SCP components (VTAM, Nucleus, IOCS, etc). It is also likely that common industry code functions will be packaged into this distributed ISS support. It will be necessary to provide interfaces from SCP to SCP (ISSS) Code to make the ISSS function completely usable.

A named data set is a record level "private" file established and maintained by a given application. The application may be implemented as a distributed set of programs sharing the data. Named data sets will contain "operational" data which is essentially temporary in nature and will be replaced or modified at regular intervals. The named data sets may be created and deleted as needed by the application. It is essential that the OPEN, CLOSE, READ, WRITE functions provide minimum response times and overhead to the user. Named data sets may be either main storage or secondary storage resident. In addition, where the host provides data base support, the application programs (host or subsystem) should have access to the data base. A data base is a multi-volume file structured on a field basis and shared by many applications generally through a data base manager function. The data base will contain data which is essentially permanent in nature--tool, maintenance, and production histories; Engineering information, et cetera. Figure 3B indicates the seven possible targets for data interchange with an application program in the host. An application in the subsystem should be able to exchange data with a host application, host or subsystem named data sets, host data base, or another application in the subsystem.

ata Base

To support applications of the type which transmit data from the subsystem to the host (or vice versa) to be manipulated (trended, reduced, etc.) an option should be allowed to request execution of a named program. This option to the data transmission request should be allowed regardless of the target data location.

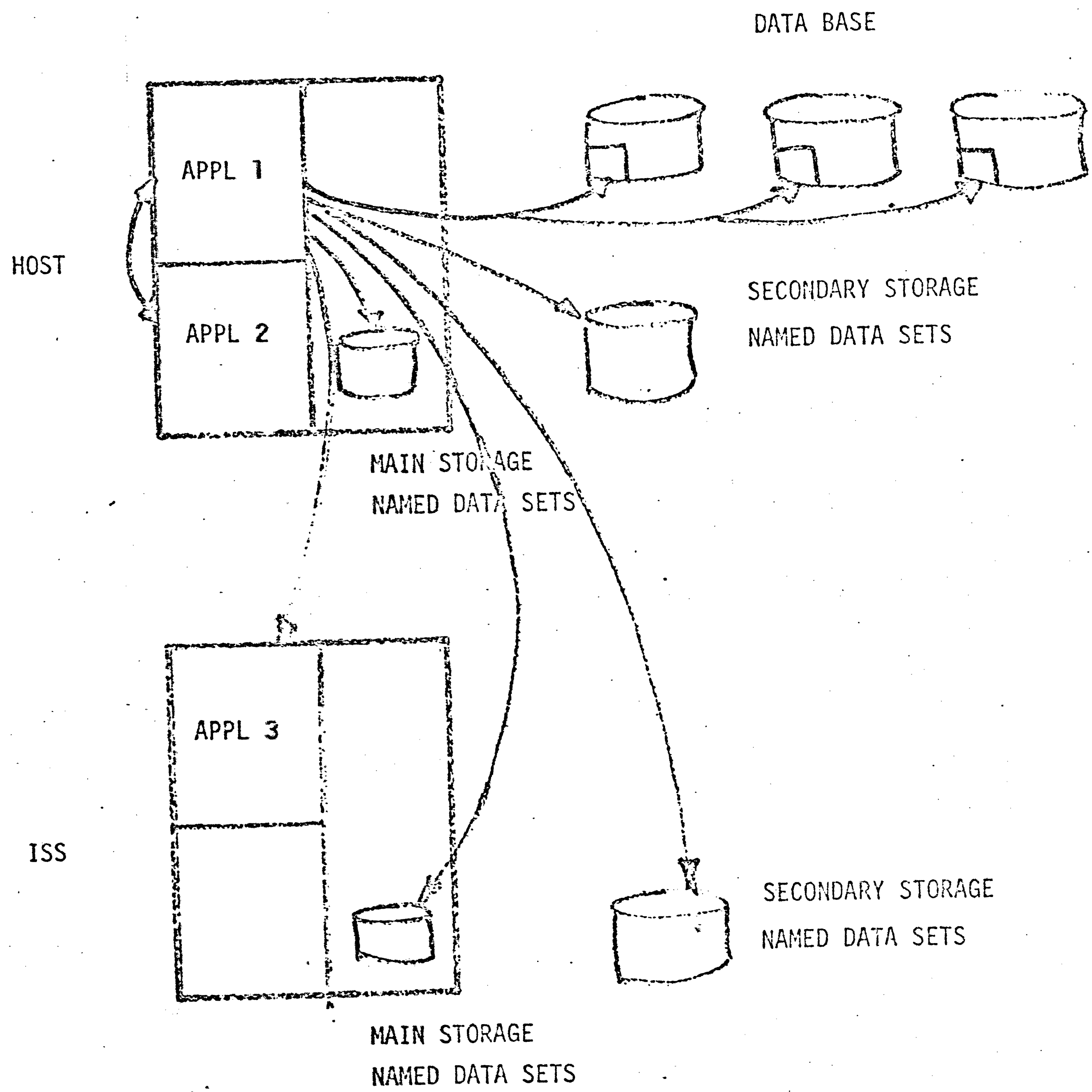

FIGURE 3B

DATA INTERCHANGE

3.5 DATA SET/PROGRAM LIBRARY MAINTENANCE

In any practical set of hierarchical system/subsystem structures, data sets and program libraries can be expected at any combination of levels. Data developed in a subsystem will need to be stored on host files. Programs prepared for subsystems on the host will need to be stored in libraries on the host and on the subsystem. Programs prepared on one subsystem for use by another subsystem will need to be stored at the host and at the target subsystem. Temporary operational data sets on the system or subsystem will need to be created, updated, then deleted from the subsystem or system.

Complete flexibility in the creation, deletion, maintenance of data sets and program libraries on one system/subsystem from another subsystem/system will be required to meet anticipated sensor base and terminal subsystem application requirements. The functions listed in section 2.3.5 must be provided. Additionally, a mechanism will be required to allow the system/subsystem to determine what data sets and libraries exist on the other subsystem/system.

3.6 CONNECTION MODES/SPEEDS

As shown in Section 2.1, support is required for both common carrier and high performance in-plant data links. In many establishments, the low speed terminal requirement will co-exist with a high-speed sensor base requirement on the same physical floor. The ISS must provide for both common carrier and in-plant line speeds and disciplines. It must also be capable of handling both simultaneously in a mixed environment. For example, connection of low speed buffered terminals to a high speed line is not an adequate solution for mixed environments where some line speeds will be determined by physical terminal locations that force common carrier connection. The ISS support of various connection modes and speeds must be made completely transparent to the user.

To allow the freedom to implement low-cost solutions to application requirements, the ISS must allow both integrated and standalone control units between the host and the hierarchy. Consideration should be provided to allow non-intelligent control units for hierarchies which are simple "one-on-one" or "tree" structures.

3.6 (continued)

In support of large volume high performance configurations the ISS must provide performance (response time, overhead, messages/second) that are consistent with transmission time on a high speed (250 Kbyte) line. Improvements in line speed should yield equivalent improvements in system performance to the user.

3.7 OPERATIONAL REQUIREMENTS

Section 2.3.7 is not an exhaustive list but rather an indicator of operational requirements of intelligent system/subsystem structures. As these configurations are applied more and more to automating businesses, as more and more manual procedures are put on line, the need for 24 hour 7 day operation increases dramatically. Intelligent Subsystem support must provide for RAS support applicable to this environment. It will be necessary to perform on line diagnosis of the component units in the hierarchy or network without stopping normal operations. It must be possible to request tests from either the system or the subsystem. In event of failure every effortmust be made to provide graceful degradation or fail soft capability. ISSS should make provision to enable specific industry products to implement RAS support to meet their requirements.

There is a rapidly accelerating acceptance of the minicomputer and other forms of low-cast intelligence to perform complex subsystem functions. Many of these will be structured without mechanical I/O and connected to a host for cost and maintainability reasons. Systems are being installed today with a separate subsystem for each application even though a single subsystem could handle all applications. Additional minicomputers (without DPIO) connected to a host by a datalink are less expensive than the cables necessary to bring all sensor inputs to a central location. These trends result in many of the listed requirements. Multiple, high volume subsystems on a host will become more common. To minimize the cost and size of these subsystems the host must provide programs and data, and access to DPIO at speeds consistent with those the subsystem could provide with local I/O.

3.7 (continued)

Many of the subsystems will be unattended with the normal operator function having been moved back to the host system. The host must provide necessary IPL, Recovery, diagnostic, et cetera functions.

As more and more of the application is moved into subsystem intelligence, the user will require high level language access to the ISS functions.

Because hierarchical structures are installed in stages, some applications will always be run on systems outside the hierarchical or network structure. Often these "external" systems generate data on a batch basis that needs to be introduced into the hierarchy. Because of this, the user will need to replace data sets and libraries with updated versions. In cases where the hierarchy is running 24 hours a day or where business timing demands replacement while in execution, this replacement must be made dynamically. There is a requirement to remove a physical file from the system and replace it with an updated version without stopping the system execution of the application.

In all hierarchical system/subsystem structures, the user will need to assure the security of his data and the operation of his system. This need can be expected to require different implementations for different users. For example, a user with terminal originated access to his data base may insist that all transactions be verified by an application in the host before access is granted. Conversely a user who has truly distributed an application between a host and subsystem may want subsystem access to the data base without any host verification to gain performance. The verification process will have been moved outboard to the subsystem. ISS must provide the required security function with flexibility to meet user requirements.

4.0 DEPENDENCIES

4.1 SNA/IN-PLANT

Development of an acceptable strategy/solution to the in-plant communications requirement is key to ISS support. SNA today doesn't address the need, outside of allowing for "device attachment", thus there is no planned VTAM, SSS, or DB/DC support of in-plant communications. Several approaches to in-plant communications exist or are planned ala the SBCU, 2790 loop, 3272, RTS-III loop, etc. This technology area must be stabilized through extension of the Standard Network Architecture (or equivalent), and properly supported to meet the needs of ISS support. (See References 3, 5, and 6).

4.2 DB/DC

There are three dependencies associated with DB/DC:

(1) Implementation of many of the ISS support functions will undoubtedly fall within the DB/DC mission, thus satisfying the requirements outlined in this document will require their response.

(2) The major DB/DC subsystems (IMS and CICS) must be merged to achieve maximum customer acceptance of the DB/DC parts of the ISS support. Multiple implementations of the DB/DC ISS functions will both hamper our ability to effectively support some system environments, and deter customer plans due to growth/migration concerns.

(3) Maximum penetration of the ISS host potential depends on migration of most of the DB/DC functions back to Type-1/SCP. That accomplishes both the minimizing of customer concerns about future commitment and strategies, and the minimizing of "front-end-load"--the requirement for the customer to pay for a Program Product in order to effectively apply the fundamental system architecture, even for a single ISS.

4.3 OPERATING SYSTEMS

The providing of a responsive host real time multi-tasking environment for user applications may well be one of the more critical requirements for implementation of distributed systems. That environment must provide for highly efficient task initiation and switching, since its usage will be primarily characterized by a large number of short duration "transaction processing" tasks, as opposed to a few long running jobs handling multiple transactions. The "Region Manager" must allow for multiple resident user and system tasks, with an open-ended library of callable tasks. It must also provide for multi-tasking, and allow implementation of single usage, multiple copy, re-usable, and re-entrant tasks.

Since the real time region will be used primarily for handling short duration tasks, that region must be viewed as only one part of the total on-line user system requirement. The operating system must allow interaction between that environment and the remainder of the system, providing for the following inter-region/partition functions:

(a) Start of jobs.

(b) Exchange of data.

(c) Program synchronization (event recognition).

For information on prior implementations, see References 2, 4, and 11.

5.0 FS IMPLICATIONS

FS is being defined with significant improvements over 370 in capability to function as a host to intelligent subsystems. A corporate direction toward intelligent communication subsystems and products has been established and augmented by the definition of the Standard Network Architecture (Reference 5) and the UC based terminal products. FS Sensor Base Objectives and Strategy (Reference 6) specify support of hierarchical structures in a layered approach to meet Sensor Base market requirements. In summary, FS will be heavily oriented to support of intelligent subsystems and application solutions based on hierarchical structures.

It is apparent that a structured Intelligent Subsystem Support package on S/370 will both meet the needs of a market that exists now and establish the correct IBM and user direction to enhance FS acceptance.

APPENDIX 6.1

MARKETING
GENERAL SYSTEMS DIV.

DEC 29 10 01 AM '72

IBM CONFIDENTIAL

SDD Harrison
December 21, 1972

copy to: J. B. Mayo
954/005-3 Rm. 313
Poughkeepsie

from: J. E Stuehler

Memorandum to: Mr. R. A. Hase
Mr. C. R. White

Subject: GSD/SDD Issue - Plan for ISS Support for System/7 on 370

Reference: Boca Raton Meeting - December 13, 1972
Action Plan Item #1

Harrison Meeting - December 19 and 20, 1972

The attached plan is the result of the work we did in Harrison and represents what I believe is a proper plan to accomplish the definition of the host function required to support System/7 as an intelligent sub-system attached to the 370. In addition, it should allow us to clarify the attachment and support via SBCU/ISBCU and lead to clarifying the proper host attach strategy.

The plan identifies dates, actions and (where appropriate and available) names. The GSD and Sensor Based Project Office (R. Hase) actions are acknowledged to be committed to this plan. Until SDD has finalized its activity regarding the establishment of a proper focal point for this activity, please use my name as the coordinating point for those activities requiring SDD involvement.

May I have your response by December 27, 1972.

R. C. Loerch

RCL:kjc
Attachment

cc: Mr. M. A. Belsky
Mr. J. Stueller
Mr. J. Weber

PHASE I - OBJECTIVE: S/7 Host Requirements Vs. UC Host Requirements & Identify Mismatches

(1) Identify Task Force Members

(a) SPD/SBPO - R.A.H. (1/2)/J. W. Marks
(b) GSD/Host - C.R.W. (1/2)/Dick Wingo
(c) SDD/UC Planning
(d) SDD/Strategy

Complete 12/22/72

(2) Identify Interfaces for Requirements/Existing Plan Input

(a) DPD Industry & SB Product Marketing
(b) SDD Industry - R. F. Bettendorf
(c) SDD Terminals - Bob Sippel/John Weber
ROCKET 3/Host - Howard Liverance
SNA Arch. - Lynn Hoberecht
(d) SPD Mfg. - Gail Nevill

(3) Game Plan/Schedule

(a) Review existing documentation (DSP, UC/Batch, UC/Interactive, SNA 1/7/73, ROCKET 3 objectives/specs)
~~GSD to distribute documents to task force by 12/27~~
(b) Interactive meetings with interfaces (2.a, b, c & d) 1/17/73 - 1/24/73
(c) Phase I Output - Match/Mismatch ISS Requirements, SB vs. UC 1/31/73

PHASE II- OBJECTIVE: S/7 Support Plan & I.D. Requirements Not Met

(1) Input Requirements (Phase I) to Florac/Roland Pampel 2/1/73

(2) Review current implementation documentation - VTAM, CICS/IMS 2/9/73

(3) Identify system performance interfaces/Review complete 2/16/73 VTAM, OS/VS, DOS/VS, CICS/IMS

(4) Schedule Belsky response to Reqûirements (open)

(5) Management Review - 5 (Item 4)

IBM CONFIDENTIAL

ISSS FUNCTIONAL REQUIREMENTS TASK FORCE - ATTENDEES

D. R. Durkin	- 23L032-2 Boca Raton
J. W. Forman	- G52 Endicott
R. R. Guyette	- F646A56 Endicott
H. K. Hallman	- 541202-1 Kingston
W. E. Harrington	- 23Z032-2 Boca Raton
B. W. Landeck	- D57 Poughkeepsie
J. M. Marks	- F626A56 Endicott
W. C. Matthews	- F50060 Research Triangle Park
J. B. Mayo	- 954005 Poughkeepsie
G. R. Nevill	- 686 Harrison
J. Steuhler	- 889 Atlanta
J. L. Weber	- 676201 Kingston
R. [illegible]	- 234 [illegible]

IBM CONFIDENTIAL

Sensor Base Project Office
Department F626A56
Tie Line 252-4313

January 22, 1973

MEMORANDUM TO: Attendees

SUBJECT: ISSS Functional Requirements Task Force - Meeting in Boca Raton January 17/18, 1973

The most significant result of the ISSS task force was agreement that functional requirements for Intelligent Subsystem support are very nearly identical for both Sensor Base and Rocket Intelligent subsystems.

These requirements are documented in the attached lists submitted by B. Landeck, J. Weber, and J. Marks. The S/7 ISSS requirements charts show the commonality and the task force estimate of the degree of support provided (for UC based products) by VTAM and SSS/SNIP. Even though general support of a function is indicated there will probably be product unique support required for both S7 and specific UC based products. It is also probable that modifications as extensions to the base packages will be required when detailed development plans are defined.

Additional work is required to confirm specific implementation requirements, evaluate performance considerations, et cetera. As a first step J. Marks and B. Mayo will expand the task force lists into an ISSS Functional Requirements document the week of January 22. This document will be reviewed with key Industry Marketing and Product groups, then submitted as a common requirements statement for UC, S7 and S3 subsystems by the end of January. Subsystem development groups can use the common requirements document as a basis for defining specific subsystem implementation requirements.

I appreciate your participation in the task force. You will receive copies of the resulting document and your comments are solicited.

RAHase

R. A. Hase

cc:	R. F. Bettendorf	R016A56 Endicott
	R. E. Carty	28Q031-2 Boca Raton
	W. L. Gee	28Q031-2 Boca Raton
	V. E. Hettinger	823DPH Harrison
	R. C. Loersch	733 Harrison
	P. H. Luhrsen	R206A56 Endicott
	H. M. Morton,	748DPH Harrison
	C. R. White	28Y031-2 Boca Raton

SUBSYSTEM SUPPORT FUNCTIONS

- HAP/SAP COMMUNICATIONS
- HAP/SDS ACCESS
- SAP/HDS ACCESS
- SAP LOADING FROM HOST
- SUBSYSTEM IPL FROM HOST
- SAP INITIALIZATION FROM HOST
- SAP POSTING HOST
- HAP POSTING SUBSYSTEM
- HAP INITIALIZATION FROM SUBSYSTEM
- SUBSYSTEM MAINTENANCE

NOTE:

HAP	-	HOST APPLICATION PROGRAM
SAP	-	SUBSYSTEM APPLICATION PROGRAM
HDS	-	HOST DATA SET
SDS	-	SUBSYSTEM DATA SET

IBM CONFIDENTIAL

J. WEBER

UC ISS REQUIREMENTS

- O DATA TRANSFER - ISS TO HOST DATA SET
- O INITIATE JOB OR TASK ON HOST FROM ISS
- O ISS REQUEST DATA FROM HOST
- O ISS REQUEST PROGRAM FROM HOST
- O DATA TRANSFER - HOST TO ISSS
- O PROGRAM TRANSFER - HOST TO ISS
- O EXECUTE ISS PROGRAM FROM HOST
- O IPL FROM HOST

S/7 ISSS REQUIREMENTS

	REQUIRE			
	S7	UC	SNA/VTAM	SS/SE
(1) INITIAL PROGRAM LOAD SUBSYSTEM				
* AT REQUEST OF HOST APPL PROGRAM	X	X	FO	X
* AT REQUEST OF UNINITIALIZED SUBSYSTEM	X	X	X	X
* FROM HOST LIBRARY VIA TRANSMISSION LINE	X	X	X	X
* FROM SUBSYSTEM LIBRARY	X	X	–	–
* CONTROLLED TERMINATION OF RUNNING SUBSYSTEM	X	X	?	NA
* TO A DIFFERENT SUBSYSTEM (?)	X	X	X	X
(2) PROGRAM LOAD OF SUBSYSTEM				
* AT REQUEST OF HOST APPL PROGRAM	X	X	FO	X
* AT REQUEST OF SUBSYSTEM APPL PROGRAM	X	X	X	X
* FROM HOST LIBRARY VIA TRANSMISSION LINE	X	X	X	X
* FROM SUBSYSTEM LIBRARY	X	X	–	–
* WITH OR WITHOUT EXECUTION	X	X		
* WITH OR WITHOUT ASSOCIATED DATA TRANSMISSION	X	O	–	–
* TO A DIFFERENT SUBSYSTEM (?)	X	X	X	X
(3) REQUEST EXECUTION OF SPECIFIC PROGRAM				
* IN HOST FROM SUBSYSTEM APPL PROGRAM	X	X	NA	N
* IN SUBSYSTEM FROM HOST APPL PROGRAM	X	X	NA	N
* IN ANOTHER SUBSYSTEM FROM A SUBSYSTEM (?)	X	X	NA	N
* WITH OR WITHOUT ASSOCIATED DATA TRANSMISSION	X	X	NA	N
* WITH OR WITHOUT KNOWLEDGE OF PAGING, LIBRARY REFERENCE, ETC.	X	Y	NA	N
* WITHOUT PARTITION/REGION RESTRAINTS	X	X	NA	N

S/7 ISSS REQUIREMENTS

	REQUIRE S7	REQUIRE UC	SNA/[illegible]	[illegible]
(4) REQUEST DATA RETRIEVAL				
* FROM EITHER HOST APPL, SUBSYSTEM APPL PROGS	X	X	NA	NA
* FROM NAMED DATA SETS.	X	X	NA	NA
* FROM HOST DB (ASSUMING NO DISTRIBUTED DB)	X	O	NA	NA
* FROM NAMED APPLICATION PROGRAMS	X	X	NA	NA
* WITH OPTIONAL REQUEST FOR PROGRAM EXECUTION	X	O	NA	NA
(5) TRANSMIT DATA.				
* AT REQUEST OF HOST OR SUBSYSTEM APPL PROGRAMS	X	X	NA	NA
* TO NAMED DATA SETS	X	X	NA	NA
* TO HOST DB	X	O	NA	NA
* TO NAMED APPLICATION PROGRAMS	X	X	NA	NA
* WITH OPTIONAL REQUEST FOR APPL PROG EXECUTION	X	O	NA	NA
(6) CREATE/DELETE/MAINTAIN NAMED DATA SETS				
* ON HOST AT REQUEST OF SUBSYSTEM APPLICATION	X	X		
* ON SUBSYSTEM AT REQUEST OF HOST APPLICATION	X	X		
* IN MAIN STG, SECONDARY STG	X	X		
(7) CREATE/DELETE/MAINTAIN PROGRAM LIBRARY				
* ON HOST AT REQUEST OF SUBSYSTEM APPLICATION	X	X	X	
* ON SUBSYSTEM AT REQUEST OF HOST APPLICATION	X	X	F	

S/7 ISSS REQUIREMENTS

(8) SUPPORT FULL RANGE OF CONNECTION MODES/SPEEDS

	REQUIRED S7	REQUIRED VC	[illegible]	[illegible]
* COMMON CARRIER AND INPLANT LINE DISCIPLINES	X	O	NA	NA
* INTEGRATED AND STANDALONE CONTROL UNITS	X	O	NA	NA
* RESPONSIVENESS COMMENSURATE WITH LINE SPEED	X	O	NA	NA

(9) OPERATIONAL AND RAS CHARACTERISTICS

	REQUIRED S7	REQUIRED VC	[illegible]	[illegible]
* 24 HOUR, 7 DAY ENVIRONMENT	X	X		
* UNATTENDED SUBSYSTEM OPERATION	X	X		
* LIBRARY AND DATA FILE REPLACEMENT (DYNAMIC) WITH NEW VERSION OF SAME DATA SET.	X	X		
* SECURITY: UNAUTHORIZED ACCESS CONTROL	X			
* HIGH VOLUME, MULTIPLE SUBSYSTEM LOAD ON HOST	X	X		
* H. P. TARGET: EQUAL OR FASTER THAN SUBSYSTEM DISK	X	X		
* USER INTERFACE CONSISTENT WITH HLL	X	X		

X = REQUIRED OR PROVIDED

O = NOT REQUIRED

? = UNKNOWN

FO = FOLLOW ON PLAN

NA: NOT APPLICABLE TO THIS COMPONENT

– = APPLICABLE BUT NOT PROVIDED

IBM CONFIDENTIAL

In the requirements list applicable to SSS/SNIP, "covered" means that the function is generally supported. It must be understood that there will most probably be GSD product unique support developed by GSD for integration into the base SSS/SNIP package.

There may also be some modifications or extensions to base SSS/SNIP required when detailed development plans are defined. This development work would be done directly by the SDD SSS/SNIP development team but is not currently funded.

In any case, additional, detailed level work is required to confirm these judgments, evaluate performance considerations, etc. This will proceed in the conventional manner i. e. SSS/SNIP will be made available to GSD. It is GSD's responsibility to review this base support and determine if it is fully responsive to GSD product requirements, and to identify and justify modifications or extensions that are required. The SSS/SNIP developers will assist GSD in defining these requirements and of course will be responsible for all design.

W. C. Matthews