DATE DUE

| | | | |
|---|---|---|---|
| | | | |
| Jan 26 83 | | | |
| 8/8/83 | | | |
| ~~8/12/83~~ | | | |
| 12-20-85 | | | |
| 9/2/98 -NIST | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

S. Department
Commerce

tional Bureau
Standards

# Computer Science and Technology

NBS Special Publication 500-71

# Remote Record Access: Requirements, Implementation and Analysis

# NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

**THE NATIONAL MEASUREMENT LABORATORY** provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities[2] — Radiation Research — Thermodynamics and Molecular Science — Analytical Chemistry — Materials Science.

**THE NATIONAL ENGINEERING LABORATORY** provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering[2] — Mechanical Engineering and Process Technology[2] — Building Technology — Fire Research — Consumer Product Technology — Field Methods.

**THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY** conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

Programming Science and Technology — Computer Systems Engineering.

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted;
mailing address Washington, DC 20234.
[2]Some divisions within the center are located at Boulder, CO 80303.

# Computer Science and Technology

NBS Special Publication 500-71

# Remote Record Access: Requirements, Implementation and Analysis

Helen M. Wood
Stephen R. Kimbleton

Center for Computer Systems Engineering
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, DC 20234

**U.S. DEPARTMENT OF COMMERCE**
**Philip M. Klutznick, Secretary**

Luther H. Hodges, Jr., Deputy Secretary

Jordan J. Baruch, Assistant Secretary for Productivity, Technology and Innovation

**National Bureau of Standards**
Ernest Ambler, Director

Issued December 1980

## Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer unitilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-71
Nat. Bur. Stand. (U.S.), Spec. Publ. 500-71, 41 pages (Dec. 1980)
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1980

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Price $2.25
(Add 25 percent for other than U.S. mailing).

# TABLE OF CONTENTS

# REMOTE RECORD ACCESS: REQUIREMENTS, IMPLEMENTATION AND ANALYSIS*

Helen M. Wood
Stephen R. Kimbleton

A key support component for network-wide data sharing is the ability of a process to access remotely stored data at runtime. In order for the accessed data to be useful, a means of overcoming differences in data representation and format is necessary. Such a capability is termed remote record access. This paper identifies some of the problems inherent in the sharing of data among dissimilar computer and data systems. Implementation issues and alternatives are presented, followed by a description of XRRA, the Experimental Remote Record Access component which has been implemented as part of the Experimental Network Operating System (XNOS) at the National Bureau of Standards.

Key words: computer networking; data conversion; data translation; data transformation; data transfer; network operating systems.

## 1. INTRODUCTION

The emergence of computer networks from the research stage to the production environment has been accompanied by a growing need to buffer the network user from the components of the network. Such a buffer would mask the differences between computer systems (hosts) on a network, thus allowing network users to spend less time learning the idiosyncracies of each system and more time utilizing network services. Network Operating Systems (NOSs) [KIMBS 76, 78], [FORSH 78] are intended to provide this type of buffer by supporting and simplifying access to existing services by simplifying interaction among systems and between systems and users.

Crucial to the realization of NOS objectives are the abilities to (1) exchange data between cooperating (but not necessarily colocated) processes, and (2) preserve the meaning, and hense the usefulness, of that data as it is exchanged between possibly heterogeneous computer systems. Traditionally, when it was known that a program on one computer system would require data from another, a decision was made to colocate the program and data on whichever system would require the least effort and expense. Although for certain high bandwidth applications, colocation may still be preferrable, the increasing size and complexity of programs, files, and data bases, coupled with the often rapid response time requirements for information, make such an approach insufficient. The ability for a process on one machine to access and make use of data on another at run-time thus has become a prerequisite for realizing the full potential of computer networking. Such a capability is termed Remote Record Access (RRA).

---

*Certain commercial products are identified in this report in order to adequately specify the procedure being described. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the product identified is necessarily the best available for the purpose. Partial funding for this work was provided by the U.S. Air Force Rome Air Develoment Center under Contract No. F 30602-77-0066.

This paper discusses the issues and alternatives related to the implementation of a remote record access capability. The remainder of Section 1 identifies goals of and solution requirements for a remote record access facility. Section 2 considers the data conversion problem in depth, and includes descriptions of related efforts. Section 3 identifies various structural considerations including the functional and information requirements and architectural alternatives involved in implementing an RRA capability. The NBS implementation of the Experimental Remote Record Access component (XRRA) is then presented, followed by a discussion of RRA in the context of higher-level, communications protocols.

## 1.1 RRA Objectives

A basic design objective for a RRA service is providing process independence from data location and originating format. It is envisioned that a RRA capability would be of most use in support of network access to data base management systems (DBMSs) and exception reporting systems (i.e., low bandwidth applications, as previously mentioned).

Location transparency seems a fairly straightforward, bounded problem primarily requiring a source of knowledge about network-wide resources (e.g., a network resource directory). Data format independence, however, may not be nearly so feasible if the range of support is not carefully specified.

When discussing protocols for data sharing, Kimbleton [KIMBS 78] noted that data transfer protocols can be distinguished by three levels of difficulty, depending on whether the block of data is generated by: i) a given data element type (e.g., characters), ii) a pointer free structure (e.g., a COBOL record), or iii) a structure containing pointers.

Case (i) is clearly feasible, as this is the case supported by the ARPANET File Transfer Protocol (FTP). Case (ii), however, is significantly more difficult. A description of the structure's graph is required, along with an identification of the structure's data elements, the mapping between structures, and complex programs to manipulate this information. The examples of real and character data representations, shown in Figures 1-1 and 1-2, are indicative of the complexity of the problem at just the data-type level.

Supporting data independence for structures containing pointers (case iii) is likely to prove extremely difficult. This is primarily because of the architectural dependence which can exist between the interpretation of the pointer and its representation. It should be noted during this discussion, that if host access methods are used to retrieve data, then any physical incompatibilities due to secondary storage formats (e.g., blocking factor) need not be considered.

This approach is therefore compatible with the concept of "protocol" as set forth bv Crocker [CROCS 72]:

> When we have two processes facing each other across some communication link, the protocol is the set of their agreements on the format and relative timing of messages to be exchanged. When we speak of a protocol, there is usually an important goal to be fulfilled. Although any set of agreements between cooperating (i.e., communicating) processes is a protocol, the protocols of interest are those which are constructed for general application by a large population of processes in solving a large class of problems.

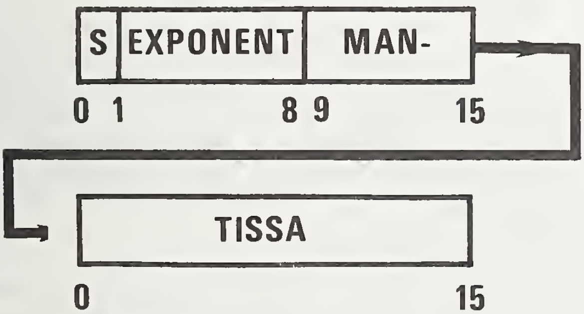
## H6180


- EXPONENT 2'S COMPLEMENT
- MANTISSA 2'S COMPLEMENT

## DECSYSTEM-10


- EXPONENT IS 1'S COMPLEMENT, EXCESS $200_8$ CODE
- MANTISSA 2'S COMPLEMENT
- NEGATIVE MANTISSA FORCES NEGATIVE EXPONENT
- MANTISSA SIGN IS BIT 0.

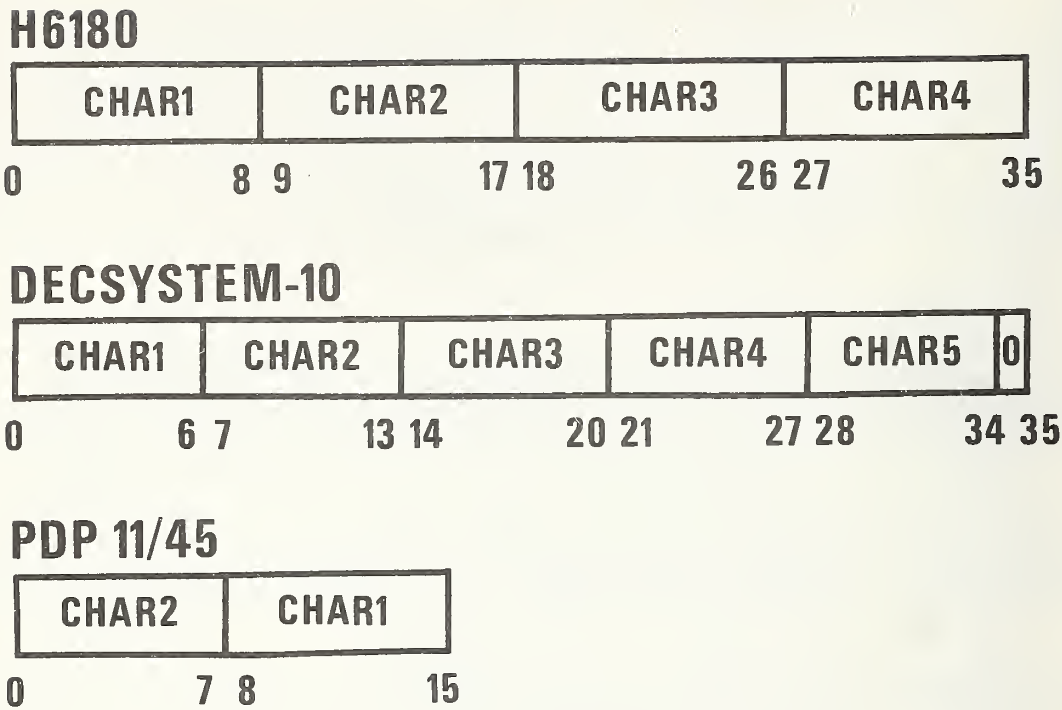
## PDP 11/45




- EXPONENT IN EXCESS $200_8$ NOTATION
- MANTISSA IN SIGN-MAGNITUTE NOTATION
- MOST SIGNIFICANT BIT OF MANTISSA NOT STORED.

**FIGURE 1-1:**

**FLOATING POINT REPRESENTATIONS OF REAL DATA**




## FIGURE 1-2:
## CHARACTER REPRESENTATIONS

The important point here, besides the generally acceptable definition of protocol, is that such tools are for "general application" by a "large population of processes" which are used in solving a "large class" of problems. Since RRA has many of the characteristics of a protocol (cf. Section 6), an approach to RRA which emphasizes breadth rather than depth (in the data conversion area) seems to be the proper alternative. Therefore, based on the above considerations, it seems desirable to confine the scope of an RRA capability to cases (i) and (ii) in the context illustrated in Figure 1-3.

Among other desirable characteristics of a RRA facility are flexibility, expandability, minimal host overhead, minimal transmission overhead, and reliability. Clearly, all of these cannot be achieved in an absolute sense in any one implementation. The development of a RRA prototype can, however, provide a wealth of substantive information that can assist in evaluating the costs and benefits of supporting such capabilities in a specific applications environment. Furthermore, such an effort can assist in the identification and development of appropriate standards for the exchange of structured data in distributed systems. For these reasons, the Experimental Network Operating System (XNOS), developed at NBS, has been utilized in exploring the basic issues in promoting more effective sharing of network accessible resources [KIMBS 78].

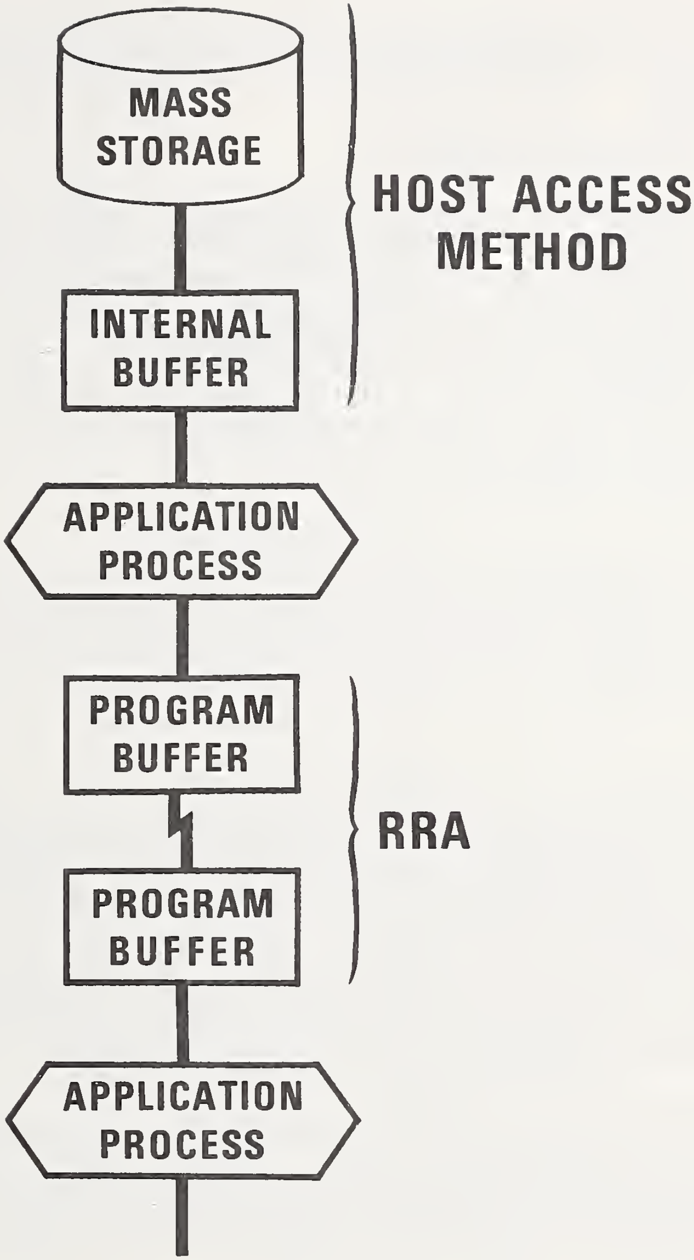



FIGURE 1-3:
RRA SCOPE

While the remainder of this paper will discuss RRA within the context of the NBS XNOS implementation, it should be noted that the functionality of the solution approach applies to the general class of NOSs represented by the NBS system.

### 1.2 Solution Requirements

In order to provide a remote record access capability, the desired data must be (i) located, (ii) accessed and (iii) any data representation incompatibilities must be resolved. The first requirement involves the specification of the host, user account (e.g., directory), file, and specific record (e.g., via access key) desired. To satisfy the second need a selection process must be available to service the request (e.g., a user program or DBMS). The support mechanisms needed to intercept a program's request for data, activate a process on the host maintaining the data to retrieve the desired record, and return the translated/transformed record to the requesting process must be built upon a protocol which supports network interprocess communication (IPC). Meeting the last specification requires sufficient information to describe the data formats, representations, and the mapping between formats, plus a transformation process to effect the data mapping.

Network Operating Systems provide a useful collection of many of the mechanisms needed to implement a RRA mechanism. Initially we assume a NOS environment as described by Kimbleton [KIMBS 76, 78]. NOSs are commonly viewed as the means for masking system differences from users. The functional objective of a NOS is to support and simplify access to existing services and to expedite the construction and subsequent accessing of new services by simplifying interaction among systems and between systems and users.

A major design goal for implementing a NOS on an existing computer network is that the NOS is transparent to the participating host systems. This goal is achievable through a consolidation of NOS support functions into a Network Interface Machine (NIM), as suggested by Kimbleton [KIMBS 76]. The NIM is, in fact, a focal point for user-system and system-system interactions. It serves, among other things, as a translator for commands (e.g., MOVE <file>, DELETE <file>), a transformer for data flowing between network processes, and a source of knowledge of network resources (e.g., maintains a network-wide file directory). The first role provides the NOS user with a standardized view of network resources by supporting a common command language for all participating hosts [FITZM 78]. The second role is actually that of the RRA component.

NBS developed XNOS to demonstrate the feasibility of such general purpose NOSs and to facilitate the investigation of the capabilities and limitations inherent in such systems. Figure 1-4 illustrates the user view of the network, while Figure 1-5 identifies the current XNOS configuration. Section 2 presents an in depth look at the problem of resolving data incompatibilities. Section 3 and 4 discuss RRA in a NOS environment in some detail.

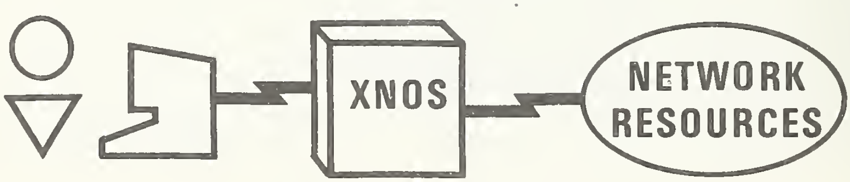
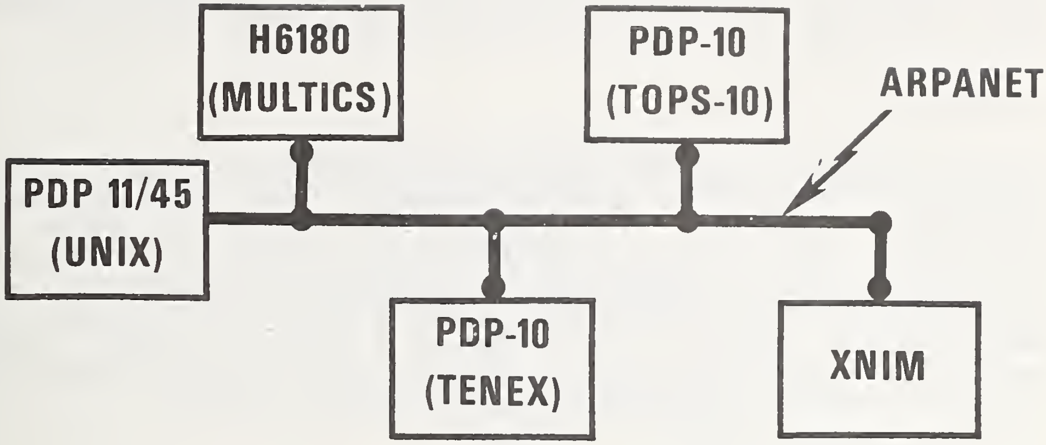



FIGURE 1-4:
USER VIEW OF NETWORK




## FIGURE 1-5:
## XNOS INITIAL CONFIGURATION

## 2. DATA CONVERSION

Incompatibilities of data representation and format are problems that preexisted computer networking. This is attributable not only to differences in data record format, but to the total lack of industry standards for the internal representation of information in computers.

The continuing need and desire to exchange computer-readable information has given rise to numerous data representation standards including for example the American Standard Code for Information Interchange (ASCII) [ANSI 1, 2], the Standard for Bibliographic Information Interchange on Magnetic Tape [ANSI 4], and the Standard Representation of Numeric Values in Character Strings for Information Interchange [ANSI 3].

In recent years, numerous efforts have been made to automate the process of transforming data. We shall now briefly describe several approaches to solving the translation/transformation problem implicit when data is shared among dissimilar hosts. It should be noted that, as presently configured, none of these systems supports run-time record translation/transformation. That is, the required support mechanisms do not currently exist to facilitate the execution-time binding of host/data names in response to a request by a program for remotely stored data. Instead, these approaches are intended to be invoked by the user directly, rather than by a process acting on the user's behalf, with the source and target data files/bases prespecified. Nonetheless, a consideration of these approaches serves to "set the stage" for identifying the issues of and requirements for the data conversion component of remote record access. (Several of these approaches are compared and contrasted in a recent internal NBS report by Fry [FRYJ 78].)

After discussing these approaches to the data conversion problem, major features of the data conversion portion of the XRRA utility are described.

### 2.1 Brute-Force Approach

In the past, "brute-force" or manual file conversion" has been the method used most often to attack the data translation problem. Thus when data in format A needed to be transformed into format B, a special purpose program was written to perform that specific transformation. Although this approach might seem acceptable for sharing data between two systems, when the number of systems increases the problem soon gets out of hand. For example, if one wished to share data between N systems, each requiring a different data format, then (N-1) translators would be needed at every host involved. Alternatively, a centralized data conversion service would have to maintain N(N-1) translators. The need for more general-purpose translation/transformation routines is obvious.

### 2.2 Generalized Approach

Within the past few years, several methods for attacking the data translation/transformation problem in a more general fashion have been suggested. Common to all of these efforts is a degree of generality and a "descriptive approach" which utilizes descriptions of the source and target data formats and a definition of the mapping to take place [BIRSE 76]. Among other factors, these generalized translation techniques can be categorized by the implementation approach adopted. For the interpretive approach a generalized processing program is developed; while a specific translation program is created for each conversion in the generative approach. Of course, some systems may involve a combination of the two.

**2.2.1 DRS.** One such conversion system, the Data Reconfiguration Service (DRS), was implemented on the ARPANET [HARSE 71, 72], [ANDEA 71], [CERFV 72]. The DRS allowed the user to specify the transformations to take place on data records (even to the bit level) through the use of a fairly complex, low-level syntax. The resulting module or "form" is essentially a "black-box" that is interjected into the communications path between user and server processes. As described in [ANDEA 71]:

> The DRS attempts to provide a notation for form definition tailored to some specifically needed instances of data reformatting. At the same time, the DRS keeps the notation and its underlying implementation within some utility range that is bounded on the lower end by a notation expressive enough to make the experimental service useful, and bounded on the upper end by a notation short of a general-purpose programming language.

The following sequence of DRS statements illustrates a form which could be used to delete 8 bits preceeding a character string [ANDEA 71]:

```
(B,,8),           /*isolate 8 bits to ignore*/

SAVE(,A,,10)      /*extract 10 ASCII characters from input stream*/

:(,E,SAVE,);      /*emit the characters in SAVE as EBCDIC characters whose
                  length defaults to the length of SAVE (i.e., 10), and advance to
                  the next rule*/
```

Such forms are used to drive a software module, called the Form Machine, which performs the specified transformation on the data stream. As shown in Figure 2-1, the DRS provides centralized transformation support.

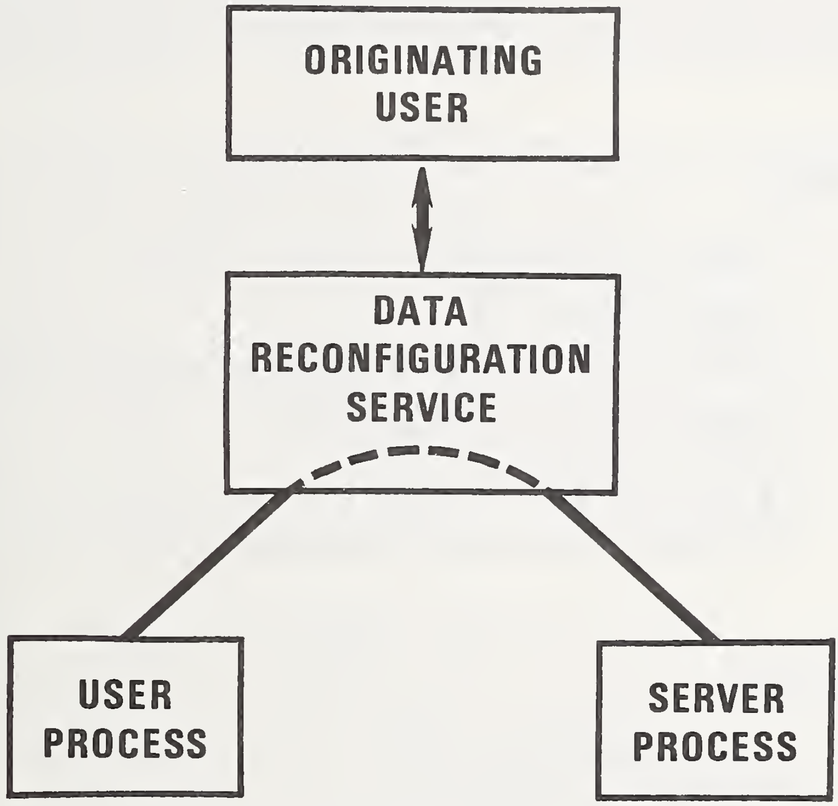



**FIGURE 2-1:**

**DATA RECONFIGURATION SERVICE**

One obvious advantage of this approach is the low data transmission overhead incurred vs. that result when a standard, perhaps character-based, format is used to communicate with the data convertor [see Section 3.1.1]. On the other hand, a clear disadvantage is the need to anticipate all needed transformations from M source formats to N target formats and provide the resulting (M x N) transformers to the DRS.

**2.2.2 DSCL.** The Data Specification and Conversion Language (DSCL), formerly entitled the File Translation Language (FTL), originated as an attempt to solve the same problem areas as DRS, but through use of a higher-level, special-purpose programming language which operates on data viewed as strings of bits. DSCL programs include a DECLARATION SECTION, in which input and output formats and representations are specified, and a

PROGRAM SECTION containing the executable statements. The flexibility provided by this higher-level language approach is evident from the example input/output declaration statements shown in Figure 2-2 [SCHNG 75A]. Here global primitives are used to define concepts such as ASCII, WORD SIZE, and CHARACTER. In addition, automatic services are provided. For example, code conversion is performed automatically whenever the declared input and output code sets of character data items taken from the input source and directed to the output set differ. Thus, in this example, the input data stream would be converted from ASCII to FIELDATA encoding.

```
INPUT

    CODE SET IS ASCII
    WORD SIZE IS 16
    DEFAULT MAPPING IS BEGIN '['=>'(';']'=>')';ALL=>'?'END
    RECORD SIZE IS VARIABLE
    EOR CHARACTER IS CR
    INTEGER REPRESENTATION IS (16,2)

OUTPUT

    CODE SET IS FIELDATA
    WORD SIZE IS 36
    RECORD SIZE IS 112 WORDS
    EOF CHARACTER IS '@'
    COMPRESSION-FLAG IS '1'B
    COMPRESSION-COUNT IS NEXT - TAB
    INTEGER REPRESENTATION IS (36,1)
```

FIGURE 2-2:
DSCL DECLARATION SECTION

It is envisioned that DSCL-like translation services could be centralized, as is the case for the DRS. Thus one machine could in effect become a network translator with DSCL used for all communications. The central machine would maintain the (M x N) translation programs required to support transformations between M source and N target data formats [SCHNG 75A,B].

As with DRS, the DSCL approach has the advantage of a low transmission overhead requirement, plus the provision of a higher-level language. However, (M x N) conversion programs are still required.

**2.2.3 SDDL.** A major area of data base research is concerned with the problem of data base transformation [FRYJ 72A, 72B, 74], [MERTA 74], [SMITD 72], [BACHM 79], and [SHUN 77]. The University of Michigan has developed an interpretive translation technique which utilizes a stored-data definition language (SDDL) to describe the source and target data bases and a translation definition language (TDL) to define restructuring transformations [FRYJ 72A, 72B, 74]. Compilers for these languages are used to produce tables of parameters which are input to a generalized translation algorithm. Although not specifically designed to support network sharing of data, if the SDDL approach were applied in a networking environment, as illustrated in Figure 2-3, it has been suggested that a single translation program could be required at each of the K hosts supporting shared data on the network. In addition, each host would also need to maintain an SDDL table describing its data and (K-1) TDL tables [BIRSE 76]. (Of course, a centralized approach could also be adopted.) Clearly, as this system was developed to solve the very difficult problem of data base conversion, it would impose a very high overhead burden when only simple data transformations are required.

**2.2.4 EXPRESS.** Two high level languages have been developed to support data translation [SHUN 76]. DEFINE is a non-procedural data descriptive language and CONVERT is a very high level, non-procedural language designed to operate on hierarchical data. In order to use these languages, input data must first be available in a normalized form. A prototype data translation system, driven by these two languages, has been developed [HOUSB 77]. Possible applications of the EXPRESS system include data base conversion and use with a centralized data base system. While able to handle highly complex transformations, just as for SDDL, this approach would impose a heavy overhead on transactions involving only simple data transfer.

### 2.3 NBS Record Translator/Transformer

The data conversion component of the NBS XRRA implementation is the Record Translator/Transformer (RTT). RTT is a generalized, non-procedural, table-driven system consisting of two modules. The first is a Record Data Translator (RDT) which performs translations between host native formats and a character-based, intermediate data format, termed Network Normal Form (NNF). The second module is the Record Transformation Routine (RTR) which performs operations on data fields within the record in order to map the incoming data to the format required by the requesting process. Tables are used to supply RTT with descriptions of the input and output data record formats, the native formats of supported network hosts (e.g., bit configuration for INTEGER data), and the mapping required to transform input records into acceptable output record formats. A more detailed description of RTT is contained in Section 4.2.

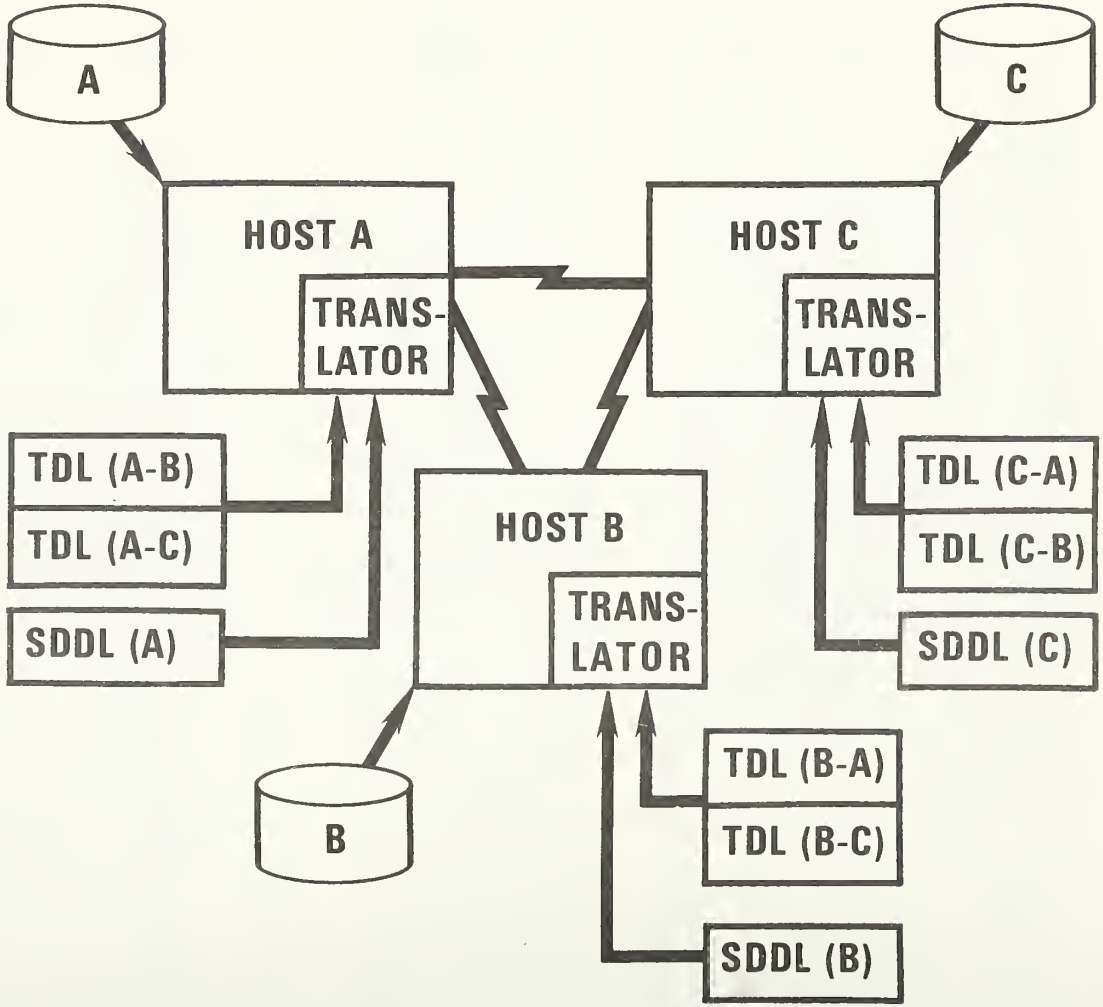



**FIGURE 2-3:**
**SDDL APPROACH IN NETWORKING ENVIRONMENT**

## 2.4 Problems of Data Transformation

Isomorphism is a natural objective in any data translation effort. If A is a record on host A which is translated into a host B representation, then we say that the translations from host A format to host B format and back again define an isomorphism provided that the original record and the record resulting from the two translations applied in succession are identical. Unfortunately, when data is exchanged by systems that support different data formats and representations, data translation problems occur which prohibit isomorphism. A common problem is loss of precision due to varying host word sizes. Other problems include format incompatibility and data type incompatibility. In the remainder of this section we shall briefly describe each of these problems. Levine examines these data translation problems in more detail [LEVIP 77].

**2.4.1 Loss of Precision.** Precision is defined as a measure of the degree of exactness with which a quantity is stated [SIPPC 72]. It is a relative term in that it is concerned with the range of values that can be represented rather than with absence of error (i.e., accuracy). Loss of precision occurs, for example, when a data item is moved from a high precision format to low precision format. Thus, attempting to represent a 32 bit integer in a host which only has 16-bit integer formats results in a loss of precision.

Precision problems cannot be avoided in a heterogeneous environment. Different word sizes and field sizes (e.g., mantissa and characteristic for real data) are the rule rather than the exception. One system may allow only single precision floating point (real) data. Another may not support floating point at all. This is not to imply that data cannot still be usefully exchanged among such systems. However, conventions must be adopted for recognizing such problems and notifying the user/server processes, as appropriate. When retention of precision is <u>essential</u>, procedures must be developed for performing the functional equivalent.

**2.4.2 Format Incompatibilities.** When describing problems with data translation, Levine notes that "..format incompatibility problems occur when data items of a particular type and in a particular format must be translated into a different format for the same type." Unlike the case of precision problems, however, format incompatibilities are strictly a function of the formatting scheme. They do not derive from the rage of values (e.g., number of bits) allowed <u>for an item</u>'s representation [LEVIP 77]. This problem is best illustrated by noting that the decimal fixed point number 0.2 cannot be exactly represented in binary. Here the transformation from decimal to binary has resulted in a change of value.

As with precision problems, format incompatibilites are unavoidable in a heterogeneous environment. Translators cannot help but introduce errors due to rounding and truncation of numeric data. Ideally, however, users will be informed of the translator's "policy" in dealing with such situations.

**2.4.3 Data Type Incompatibilities.** Data type incompatibility results when an output format does not exist to receive a given data type. One example of such a situation occurs when a process attempts to output floating point information to a terminal device (i.e., no floating point-to-character transformation has taken place). While there might be a requirement for the provision of some type of terminal handling intelligence to interface a "dumb" terminal to a "smart" network, it is still entirely possible that data type incompatibilities will occur even between other "smart" systems on a network. Therefore, some sort of error detection and recovery mechanisms must be provided to handle such cases.

# 3. STRUCTURAL CONSIDERATIONS

Although it is the consensus that, especially in a computer networking environment, the generalized approach to data conversion is preferrable to the alternative (i.e., brute-force), there is little agreement on a "standard" method for implementing such systems. Since the organization of a remote record access system has direct impact on the set of support requirements, this section will highlight some of the organizational alternatives and their related considerations/implications, after first identifying the support requirements common to all approaches.

## 3.1 Support Requirements

Based upon careful consideration of the problem, along with existing solution approaches to the several problem components, it is apparent that solving the remote record access problem requires certain easily identifiable functional and informational capabilities. Providing these capabilities, in turn, gives rise to additional needs (e.g., interprocess communication, arbitrary precision arithmetic capability, specification of a standard data format). These and other support requirements are now discussed.

**3.1.1 Functional.** The provision of a remote record access capability requires (1) a mechanism for selecting a record from the file/data base containing it, (2) a record translator to preserve meaning in transmitting the record between dissimilar hosts and (3) a record transformer to permit the alteration of record structures.

The precise mechanism which supports record selection is dependent upon capabilities existing at the host computer, including those provided by a data base management system, if any. It is assumed, from the perspective of specifying a RRA capability, that the selector process exists and is capable of retrieving a record based on utilization of a unique key, if random access techniques are employed. The keyword NEXT must be used if sequential access is being supported.

Record translation preserves the logical record structure and data element type (e.g., real, binary, logical, integer, character) and, for arithmetic data elements, precision. Clearly a record translator must know the exact format of the record to be translated, down to the data item, along with the internal format of all data types for each and every system supported.

Record transformation supports modification of both the logical structure of the record and individual data elements. Such transformations are useful in matching the information transmitted to the needs of the receiver (e.g., field reordering). They may also be utilized in controlling access to sensitive information (e.g., by omitting sensitive information from the record before transmitting it on to the requesting process). Such transformation affects the logical structure of the record through one of three basic transformation types: logical, arithmetic, or string. Among the additional transformations that may be needed are algorithms for the compression and/or decompression of textual information, as well as for field or record level encryption/decryption.

The operations currently implemented in XRRA are shown in Table 3-1. The logical transformations AND and OR generate Boolean binary strings resulting from the bit-by-bit ANDing and ORing of two successive strings. The basic arithmetic transformations +,-,/,* act as would be expected. String transformations can be quite complex as evidenced by the capabilities of string manipulation languages. Initially, a concatenation capability is supported.

| OPERATION | SYMBOL | DATA TYPE | | | | |
|---|---|---|---|---|---|---|
| | | INTEGER | REAL | CHAR | BINARY | BOOLEAN |
| ADD | + | X | X | | | |
| SUBTRACT | − | X | X | | | |
| MULTIPLY | × | X | X | | | |
| DIVIDE | / | X | X | | | |
| AND | & | | | | X | X |
| OR | I | | | | X | X |
| CONCATENATE | # | | | X | | |

## TABLE 3-1:
## XRRA TRANSFORMATION OPERATIONS

**3.1.2 Information.** As shown above, mechanisms for solving the data conversion problem require information about the physical and logical characteristics of the data. Such information must be provided explicitly since, generally speaking, strings of bits do not carry with them any indication of the data type(s) they are representing. Levine [LEVIP 77] notes that this is because "...the overwhelming majority of currently available computer systems are based on the Von Neumann philosophy for storing digital information." Consequently, the semantic meaning of bit strings is derived from the context in which they are used.

Physical characteristics are the actual bit configurations of each type of data maintained on the system. For example, floating point words on the DECSYSTEM-10 are 36 bits in length, have a sign-bit located in bit position 0, followed by an 8-bit exponent in one's-complement, excess 200 (octal) notation, which in turn is followed by a 27-bit normalized mantissa in two's-complement representation. Similar information is also required to fully describe the DECSYSTEM-10's internal representation of integer, character, logical, and Boolean data types. In XRRA, this information is maintained in the Host Representation Table (HRT). Table 3-2 illustrates HRT entries describing the format of real data for three computer systems. However, these descriptions would not be complete for all systems. For example, in the Burroughs B5500, B5600, and CDC 6000 Series computers, the radix point is at the right of the mantissa, rather than the left as for the systems in this table. Also, the IBM 360-370 Series represents the exponent in base 16, rather than base 2. (A good discussion of the plethora of data type representations can be found in [TREMJ 76].)

| | DECSYSTEM-10 | HONEYWELL 6180 | PDP-11/45 |
|---|---|---|---|
| WORD SIZE | 36 | 36 | 16 |
| MANTISSA SIGN LOCATION | 0 | 8 | 0 |
| MANTISSA BIT POSITIONS | 9-35 | 9-35 | 9-15, 0-15 |
| MANTISSA MOST SIGNIFICANT BIT STORED | YES | YES | NO |
| MANTISSA NORMALIZED | YES | YES | YES |
| MANTISSA REPRESENTION | 2'S COMP | 2'S COMP | SIGN-MAGNITUDE |
| EXPONENT SIGN LOCATION | N/A | 0 | N/A |
| EXPONENT BIT POSITIONS | 1-8 | 1-7 | 1-8 |
| EXPONENT REPRESENTATION | 1'S COMP | 2'S COMP | 1'S COMP |
| EXPONENT EXCESS CODE | 128 | 0 | 128 |

## TABLE 3-2:
## HOST REPRESENTATION TABLE FOR REAL DATA TYPE

A complete description of the organization of the data to be transferred would, on the otherhand, include such information as size of record, names of data elements (also called fields or items), and the type and size of each item. Thus, a description of a data record might look somewhat like a conventional FORTRAN format statement (e.g., 3A5,2X,5I,2X,F7.2), with names associated with each field or, more likely, resemble a COBOL data description. Whatever the form, a need exists for a language to fully describe the data - a Data Description Language (DDL).

In XRRA, a Logical Record Description (LRD) contains such information as the record length and a set of Data Element Descriptions (DEDs) which, for each data element, specify the element level (node), name, and attributes (data type and size).

**3.1.3 Standard Data Forms.** Although not a prerequisit for data translation/transformation as a practical matter the use of an intermediate "standard" or "normal" notation to represent the data is desirable. Not only would such a notation reduce the complexity of a general translation algorithm (e.g., Michigan's SDDL approach [FRYJ 74]), but for systems like DRS [ANDEA 71] or DSCL [SCHNG 75A,B] where the network translation support is centralized, the number of translation routines would be reduced from

N(N-1), for N computer/data systems, to 2N (with one algorithm defining the transformation to "normal" form, and one defining the inverse).

The use of a standard intermediate form for representing data exchanged between potentially different systems is not new. The ARPA protocols TELNET and FTP both support such a convention. The TELNET protocol provides terminal users the means of accessing remote systems as if the user were a local user of that system. Implementation of the TELNET protocol is based on a Virtual Terminal Protocol defining a network-wide set of terminal functions and character encodings. The source computer (system to which the user is logged in) maps the functions and character encodings which it uses into the corresponding VTP functions and encodings. The destination computer (remote system being accessed by the user) maps from these VTP functions and encodings into those which it supports. The ARPA File Transfer Protocol (FTP) also follows this general approach of mapping from a local host representation to a common network representation and back to a local host form. At present FTP only supports transmission of character or binary (i.e., unmapped) files.

Levine [LEVIP 77] examines the use of a standard, character-based format, i.e., characters used to represent all data types, vs. a data format consisting of a standard character set for character data and a set of more data compatible formats for other data (e.g., a non-character based format for exchanging integer data and another for real data). It is apparent that any one approach would not be optimal for all applications. Transmission and processing overhead are certainly among the major factors to be considered when choosing a standard format. For example, if large amounts of non-character data are to be transmitted in a character-based normal form, then there may be both communications bandwidth and processing overhead concerns. On the other hand, many processors currently support internal translation to ASCII (and the reverse) in order to communicate with their various terminal and other peripheral devices. Thus, looking ahead to the day when heterogeneous systems exchange structured data in a standard format, a character-based canonical form is likely to be an acceptable compromise and may even be the best general purpose alternative.

The RTT component of XRRA utilizes an ASCII-based intermediate, Network Normal Form (NNF). In this format, all data (even numeric) is represented in a character form. For example, a data field containing data of type REAL would be expressed as

$$\{+,-\}\{d_1, d_2, d_3, ...\}.\{d1, d_2, ...\}$$

where each element "dn" is a decimal digit. Binary data, on the otherhand, would be expressed as strings of the ASCII characters "0" and "1". The logical data types TRUE and FALSE appear as "*T*" and "*F*", respectively. Field delimiters may be any character that does not appear within the data fields (e.g., '|'). The following is an example XRRA record in NNF:

|WIDGITS|-03.5686| + 32.456|-15|0111001111110011|*T*|

The exchange of self-describing data, in which canonical data descriptive tags accompany the data in its travels (i.e., self-describing records), has also been suggested. A standard format for the exchange of structured data, which employs a data element tag based, data description format is now being proposed by the American National Standards Institute (ANSI) [ANSI 5]. This format is based on the ANSI Standard for Interchange of Bibliographic Information [ANSI 4] and was developed in conjunction with efforts of the Inter-Laboratory Working Group for Data Exhange (IWGDE) of the Department of Energy. It provides specifications for:

1. elemental data types -- numbers and text in code extensions

2. a set of structures -- scalars, vectors and arrays -- with associated format information as well as a higher level hierarchical structure

3. a method of naming or describing the data contained in each field or subfield.

The intent of this proposed standard is to provide the means to interchange a wide variety of information while remaining content-independent. In addition, the proposed standard utilizes the concept of a logical record which is media-independent. The ASCII character set [ANSI 1,2], is recommended as the preferred code for representing all data types, but non-ASCII coded character sets are also permitted.

Partial implementation of this proposed ANSI standard is underway within the IWGDE. Versions are planned for PL/I (IBM), DEC PDP-11, and a FORTRAN/Assembler (CDC).

**3.1.4 Process Interface.** Regardless of the implementation approach chosen, run-time support of a remote record access system requires a mutually agreed upon mechanism or protocol for interfacing user/server processes. Such a protocol, termed Interprocess Communication (IPC), provides the basic mechanism for initiating and controlling the flow of data between cooperating processes. Since processes are the only active entities within a computer system, IPC is a basic building block for supporting communication between computers.

Three increasingly sophisticated levels of interprocess communication can be identified: *job level, call/return,* and *message based.* At the job level, a basic mechanism is provided for executing a job consisting of a collection of job steps, each of which may be resident on a different system. The IPC mechanism must support initiation of a job step when the required input files are available and must also provide for migration of output files upon termination of the step. Job steps capable of concurrent execution should also be identified. Such a mechanism is provided as part of JES-2 [SIMPR 78] and as part of an Experimental Network Operating System [KIMBS 78].

*Job level* IPC only supports interaction prior to the initiation or following the termination of a job step. If one wishes to provide a run time mechanism, some attention must be given to the form of implementation. One alternative, the *call/return* based approach, allows one process to communicate with another in a manner directly analogous to subroutine calls. That is, a process issues a CALL and thereafter enters the WAIT state pending RETURN of the results.

Although the *call/return* approach is intuitively straightforward, its use in a networking environment poses certain problems reflecting uncertain delays and the likelihood of outages. In the context of an individual system, aborting a job if a system crash occurs after a subroutine *call* has been issued is unexceptional. In contrast, in a networked environment, the likelihood of communications network outages or the unavailability of a remote systems can result in exceptionally long processing delays for the *call*ing process. A better approach would be to request initiation of a remote process, continue executing, and later check to see if the desired results have been returned. This constitutes the message based approach to IPC.

*Message based* IPC provides a very flexible approach for communicating between systems. The cost is the requirement that the user program explicitly provide for transmitting and receiving messages. Although transmission might be considered to be at the same level of difficulty as supporting the *call/return* approach, message reception requires substantially more sophisticated mechanisms. This reflects the desirability of having system support in classifying messages and for permitting inspection of message queues to determine the appropriate sequence for processing. For example, it is usually desirable that a process be immediately notified whenever a remote host is down while, in contrast, handling results returned by a remote process can usually be deferred until a collection of such results are to be processed.

**3.1.5 Arithmetic Capability.** Representing and manipulating numerical data that exceeds the precision capabilities of the processing host is one of the problems that occur when attempts are made to manipulate data that is in the native form of another processor. It is not acceptable to require that the data "fit" into the word size of the processor supporting data conversion as such a requirement could result in a serious loss of information (e.g., precision loss). Representing such data in character rather than binary form (e.g., character representation of floating point data) would be one approach to the representation problem. Routines are then required that are capable of accepting variable length character (or bit) strings and performing various classes of operations on them (e.g., arithmetic, logical, string, and Boolean).

Although an arbitrary precision arithmetic capability will help prevent data precision loss during the portions of the conversion process of the source record from source host format to canonical NNF format, precision loss may still occur if the word size, for example, of the target host is less than that required to represent the data.

### 3.2 Architectural Alternatives

As discussed by Shoshani [SHOSA 72,73], there are several possible approaches to data sharing in computer networks in terms of distribution of the support components. Shoshani terms these categories: centralized, standardized, data transformation, and integrated.

**3.2.1 Centralized.** In the centralized case, network access to a DBMS may involve dealing with a specialized data base machine. Such is the case with the Computer Corporation of America's Datacomputer [MARIT 75]. In this situation programs scattered around the network interact with the Datacomputer in a common Datalanguage. This language includes facilities for describing data, creating and maintaining a data base, and the selective retrieval of items from the data base. Such centralization of DBMS services lifts from the user such tasks as learning more than one query language. However, continuing research in DBMS technology alone is an indication that it is unrealistic to assume that all DBMS-related user needs can be met by a single type of system. Thus, it is reasonable to assume that network users will require access to various DBMSs, and in fact may wish to update the data maintained by one system with information retrieved by another system having perhaps a significantly different architecture.

**3.2.2 Standardized.** In the standardized approach, the same set of data management services is implemented throughout the network. While this approach might be preferrable under certain circumstances, its implementation on pre-existing systems would be relatively difficult. That there is some movement in this direction is evidenced by the proposed data exchange formats, e.g., [ANSI 5] described above, and current efforts on the part of ANSI, the International Organization for Standardization (ISO) and others in defining a reference model for distributed systems within which standards can be established [ISO 79].

**3.2.3 Data Transformation.** The data transformation approach involves the reconfiguration of data from the form in which it is maintained on one system, directly into the form required by the system on which data processing is to take place. As Shoshani observed, "the data transformation approach can be viewed as an extension of the centralized approach to handle existing data from existing systems"[SHOSA 72]. Both the DRS and DSCL approaches discussed above are representative of this class.

**3.2.4 Integrated.** Finally, an integrated approach would involve the use of interfaces and a common language in conjunction with existing data management systems. The interfaces themselves may be physically co-located with the corresponding data management systems, or centrally located at one network location. NBS's XNOS has adopted this type of approach in its support of network data. The XNOS Experimental Network Interface Machines (XNIMs) serve as interfaces between heterogeneous computer and data base systems. A common command language is supported for file maintainence and network job execution [FITZM 78] and XRRA provides the data conversion interface for exchanging structured data. In addition, a Experimental Network Data Manager (XNDM) is now being designed and implemented at NBS to interface heterogeneous DBMSs. Users and processes will express their requests in a standard query language which the XNDM will transform into the DBMS-specific languages [KIMBS 79].

## 3.3 Design Considerations

Once an architectural approach has been selected, two major alternatives confronting a RRA designer revolve around (i) where to place the RRA support components and (ii) how to interface to other networking capabilities.

**3.3.1 Inboarding vs. Outboarding.** As illustrated in Figures 3-2a and 3-2b, RRA support components (e.g., translators, data descriptions) may be incorporated inside of existing computer systems (i.e., "inboarding") or special-purpose, perhaps dedicated, front-end or shared systems charged with these responsibilities may be developed (i.e., "outboarding"). The selection of one approach over the other must be based on an analysis of the trade-offs involved in each case.

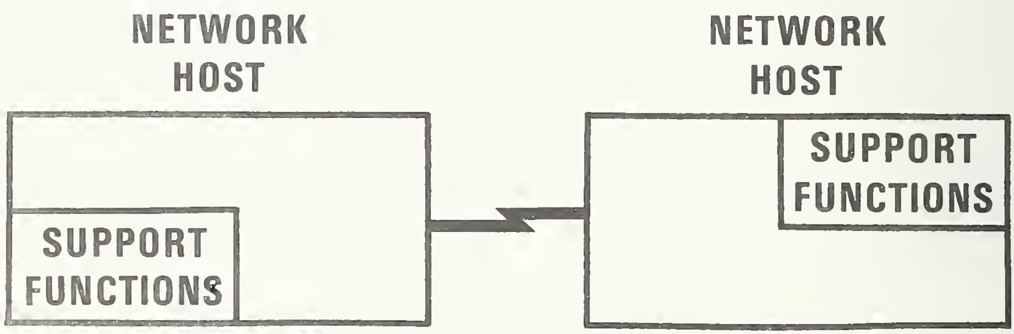
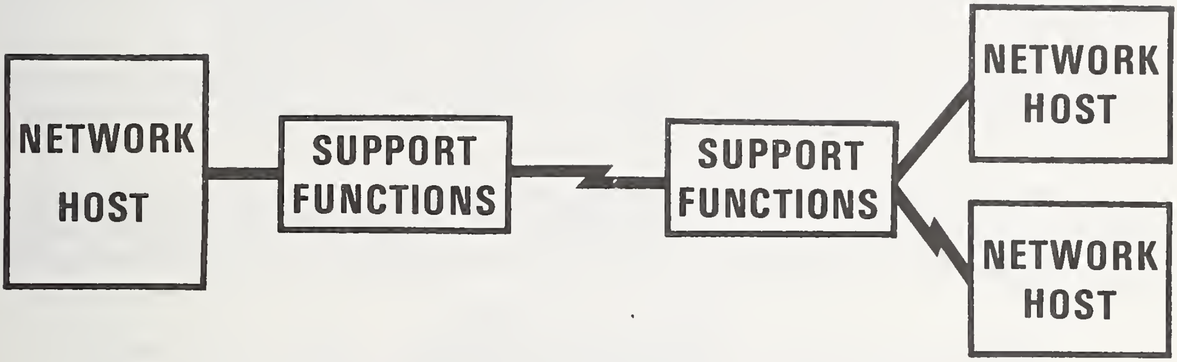



FIGURE 3-2A:

INBOARDING SUPPORT FUNCTIONS




# FIGURE 3-2B:
# OUTBOARDING SUPPORT FUNCTIONS

The XNOS implementation is an example of "outboarding" as the NOS support functions (including XRRA support) are consolidated into the XNIM. Minimal burden is placed on XNOS-participating hosts.

**3.3.2 (De)Centralization.** Whether "inboard" or "outboard," RRA support functions may be centralized (i.e., provided by one system) or distributed (i.e., spread across many systems). If "inboarded," then the decision to centralize or distribute these functions would depend on such factors as the overhead involved in implementing a general purpose translator (e.g., Michigan's SDDL) or a set of translators (e.g., DSCL) at a number of hosts. Another factor could be the utility of maintaining a centralized data base management system which is also capable of translating and transforming records to meet the needs of requesting host systems (e.g., CCA's Datacomputer).

If "outboarding" is chosen, then the demand for support system services would determine the number of systems required. For example, a network supported by XNOS might have one XNIM supporting all participating host systems (e.g., the current XNOS configuration). If demand increased sufficiently, an XNIM might be dedicated to serve one specific class of systems (e.g., Multics systems). In the most distributed case, each participating XNOS host would be served by an XNIM support system.

In the final analysis, the optimal degree of (de)centralization chosen for implementation will depend upon a combination of managerial (e.g., security, control) and physical (e.g., traffic, bandwidth) characteristics.

**3.3.3 Layering Concept.** Modularity has come to be accepted as the most desirable implementation approach for operating systems and large applications. Anderson et. al. [ANDEA 74] observe that the concept of "layering" is closely related to and in fact includes that of modularity. "Levels" are specified which define precise boundaries between different related sets of modules. At each level, the modules are implemented using the functions provided by lower levels as primitives. These levels are often referred to as "virtual machines" in operating system design.

The following principles have guided the ANSI-ISO effort to design a standard reference model of the architecture of distributed systems [BACHC 78] [DESJR 78]. They are intended for use in determining the number of layers and the best place for boundaries between layers include:

1. Create a sufficient number of layers to divide the total work into pieces small enough for easy comprehension by a single person.

2. Do not create so many layers as to complicate the system engineering task describing and integrating these layers

3. Create a boundary at a point where the services description can be small and the number of interactions across the boundary are minimized.

4. Create separate layers to handle functions which are manifestly different in the process performed or the technology involved.

5. Collect similar functions into the same layer.

To be consistent with this concept, an RRA capability should be built upon "lower-level" functions that are concerned with transporting data between computer systems. In addition, RRA should be somewhere "above" the layer in which Interprocess Communications functions reside. The exact relationship of RRA to other "higher-level" functions concerned with data exchange on an end-to-end basis (e.g., from operating system to operating system or application process to application process) remains to be fully explored. (See Section 6 for more on this problem.)

## 4. IMPLEMENTATION APPROACH

In [KIMBS 78], an overall description is given for the implementation of the NBS Experimental Network Operating System. The Experimental Remote Record Access system operates within and as an integral part of the XNOS.

This section describes in more detail the approach adopted in the XRRA implementation. The major components (e.g., functional, informational) are identified, and a detailed example of a session between two processes requiring XRRA services is presented.

## 4.1 XRRA Architecture

As illustrated in Figure 4-1, the major functional and informational components of XRRA reside on the XNIM. XRRA assumes the existence of a suitable host mechanism for retrieving a record based on utilization of a unique key, if random access techniques are being employed or, alternatively, the keyword 'NEXT' if sequential access is being used. The data conversion approach adopted in XRRA involves the use of non-procedural languages (tables) to implicitly specify the data manipulations.

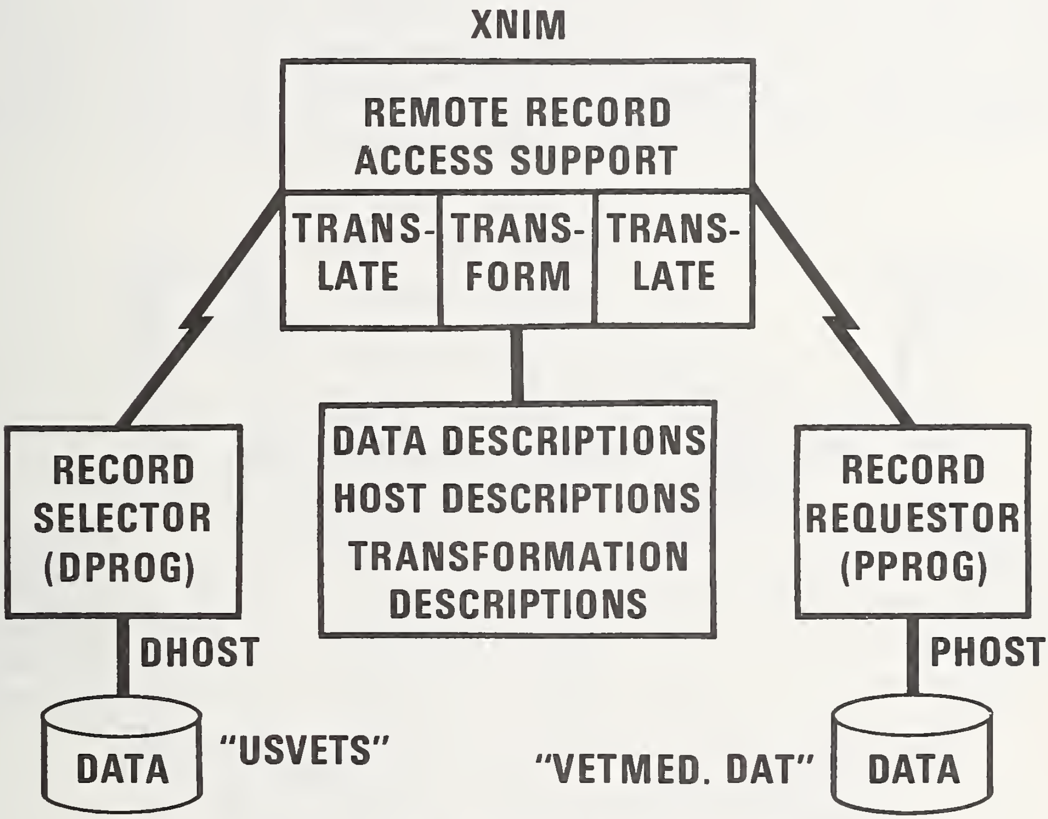



FIGURE 4-1:
XRRA COMPONENTS

The following data types are presently supported: INTEGER, REAL, LOGICAL, BINARY, and CHARACTER.

Conversions of these data types have been successfully performed on all of the systems currently supported by XNOS: Honeywell 6180 running Multics, DECSYSTEM-10 running TOPS-10 and TENEX, and Digital Equipment Corporation 11/45 supporting Bell Laboratories Unix timesharing system. ("Unix" is a Bell System Trade/Service Mark.)

### 4.2 XRRA Example

The data translation and transformation capabilities supported in providing process access to remote records can best be illustrated by following a record and its associated descriptive tables through the path from the host maintaining the data (DHOST) to the host requesting the data (PHOST). This path is shown in Figure 4-2. The DHOST in this scenario maintains a data base (USVETS) of medical records for veterans. DPROG is the data selector process available on DHOST. The Data Element Descriptions (DED) of the Logical Record Description (LRD) table for these records would then be as shown in Table 4-1.

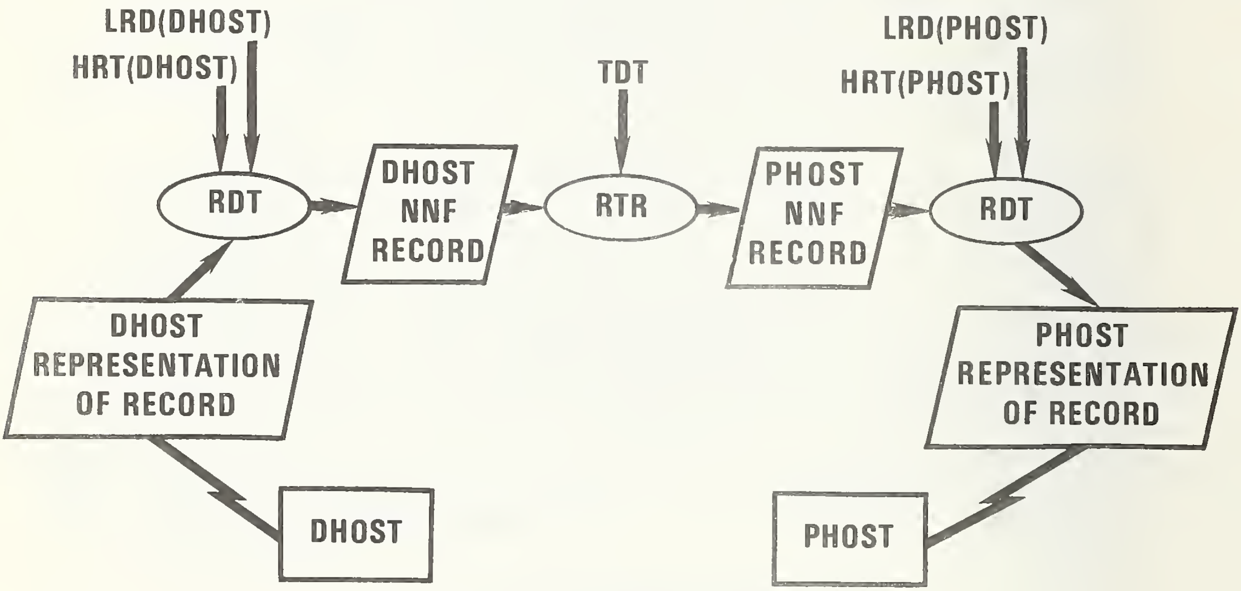



FIGURE 4-2:
DATA TRANSFER PATH

```
ID = 1,1,0,  IDNO, C(9,0)
ID = 1,2,0,  PATIENT, C(20,0)
ID = 1,3,0,  BIRTH, C(7,0)
ID = 1,4,0,  ALLERGY, B(36,0)
ID = 1,5,0,  ALLERGY TEST, B(36,0)
ID = 1,6,0,  HEIGHT, R(3,2)
ID = 1,7,0,  WEIGHT, I(3,0)
ID = 1,8,0,  SEX, C(1,0)
ID = 1,9,0,  DISEASE1, C(0,0)
ID = 1,9,1,  DISEASE1, NAME, C(15,0)
ID = 1,9,2,  DISEASE1. DATE, C(7,0)
ID = 1,9,3,  DISEASE1. MEDICATION, C(15,0)
ID = 1,10,0,  DISEASE2, C(10,0)
ID = 1,10,1,  DISEASE2. NAME, C(15,0)
ID = 1,10,2,  DISEASE2. DATE, C(7,0)
ID = 1,10,3,  DIRSEASE2. MEDICATION, C(15,0)
ID = 1,11,0,  DISEASE3, C(0,0)
ID = 1,11,1,  DISEASE3. NAME, C(15,0)
ID = 1,11,2,  DISEASE3. DATE, C(7,0)
ID = 1,11,3,  DISEASE3. MEDICATION, C(15,0)
ID = 1,12,0,  DISEASE4, C(0,0)
ID = 1,12,1,  DISEASE4. NAME, C(15,0)
ID = 1,12,3,  DISEASE4. DATE, C(7,0)
ID = 1,12,3,  DISEASE4. MEDICATION, C(15,0)
ID = 1,13,0,  PARENTS, C(0,0)
ID = 1,13,1,  PARENTS. MOTHER, C(20,0)
ID = 1,13,2,  PARENTS. FATHER, C(20,0)
ID = 1,14,0,  YRS CIVILIAN GOVT, I(2,0)
ID = 1,15,0,  YRS MILITARY, I(2,0)
ID = 1,16,0,  DISABLED VET, L(1,0)
```

## TABLE 4-1:

## DED FOR "USVETS" RECORD

PPROG is a process executing on PHOST which requires access to data that is available on DHOST. The PHOST data requirements constitute a subset of the DHOST data record. Thus, certain data fields (e.g., parent information) are not needed by the PHOST process, and in fact should not be transmitted. In addition, suppose that several other fields are to be added, or otherwise operated upon. The result of these operations would be a somewhat smaller, but in any case transformed, PHOST record as defined by the LRD shown in Table 4-2 and maintained on the XNIM.

ID = 1,1,0, SSN, C(9,0)

ID = 1,2,0, NAME, C(20,0)

ID = 1,3,0, SEX, C(1,0)

ID = 1,4,0, YRS GOVT SERVICE, I(2,0)

ID = 1,5,0, BIRTH DATE, C(7,0)

ID = 1,6,0, TESTED ALLERGIES, B(36,0)

ID = 1,7,0, SUSPECTED ALLERGIES, B(36,0)

ID = 1,8,0, WT, I(3,0)

ID = 1,9,0, HT, R(3,2)

ID = 1,10,0, DISEASE 1, C(37,0)

ID = 1, 11,0, DISEASE 2, C(37,0)

ID = 1, 12,0, DISABLED, L(1,0)

TABLE 4-2:

DED FOR "VETMED.DAT" RECORD

For each DHOST and PHOST logical record type, a Transformation Description Table (TDT) is then provided. This table, shown in Table 4-3, establishes the relationships between data items in the PHOST and DHOST records using data element names from the DED portion of the Logical Record Description Table and the operators specified in Table 3-1.

| PHOST RECORD | DHOST RECORD |
|---|---|
| SSN | = IDNO |
| NAME | = PATIENT |
| SEX | = SEX |
| YRS-GOVT-SERVICE | = YRS-CIVILIAN-GOVT & YRS-MILITARY |
| BIRTH-DATE | = BIRTH |
| TESTED-ALLERGIES | = ALLERGY & ALLERGY-TEST |
| SUSPECTED-ALLERGIES | = ALLERY \| ALLERGY.TEST |
| WT | = 2.2 × WEIGHT |
| HT | = (0.4 × HEIGHT)/12 |
| DISEASE-1 | = DISEASE3.NAME #DISEASE3.MEDICATION #DISEASE3.DATE |
| DISEASE-2 | = DISEASE4.NAME #DISEASE4.MEDICATION #DISEASE4.DATE |

## TABLE 4-3:
## EXAMPLE TRANSFORMATION DESCRIPTION TABLE

Several interesting capabilities, which result from supporting tree-structured data records, are illustrated in this example. Note that when selecting the last two disease history fields (DISEASE3 and DISEASE4) from the DHOST record for inclusion in the PHOST record, the fields, which are each composed of three subfields, are transformed via reordering and concatentation of the related subfields. The result is then assigned to the appropriate field in the PHOST record description. For example, the TDT contains the entry

"DISEASE-1 = DISEASE3.NAME # DISEASE3.MEDICATION # DISEASE3.DATE"

where '#' is the concatenation operator. This statement is functionally equivalent to the following set of statements:

DISEASE-1.NAME = DISEASE3.NAME

DISEASE-1.TREATMENT = DISEASE3.MEDICATION

DISEASE-1.OCCURRENCE = DISEASE3.DATE

Thus, one entry in the TDT describes a 3-part PHOST disease history field.

For every supported host type, necessary host-descriptive information is in the Host Representation Table (HRT) (e.g., Table 3-2). For this example, XRRA would require HRTs for the Honeywell H6180 and DEC PDP 11/45. In the initial XRRA implementation it is assumed that all data types are single precision.

The following sequence of events occur during an XRRA session:

1. A user requests activation of the PHOST process, PPROG. A user requests activation of the PHOST process, PPROG.

2. The XNIM activates the DHOST process, DPROG.

3. PPROG requests for data are intercepted and passed on to the awaiting DPROG.

4. DPROG retrieves the indicated data and returns it, in native form (i.e., binary strings) to the XNIM.

5. The XNIM then directs this data string to the Record Translation/Transformation (RTT) component of XRRA, identifying (via calling parameters) the DHOST and PHOST names, along with the LRDs which describe the data.

6. The Record Data Translation component of RTT translates the DHOST record to Network Normal Form.

7. The DHOST record in NNF is then transformed by the Record TRansformation component of RTT to meet the PHOST format requirements, as indicated by the Tranformation Record Table.

8. The resulting PHOST record in NNF is translated into PHOST native representation.

9. The PHOST record is returned to the XNOS monitor in the XNIM, which then transmits the record to the awaiting PHOST process, PPROG.

The record received from the DHOST at step 4, appears to the XNIM as a seemingly meaningless binary string represented by the character stream shown in Figure 4-3a. Figure 4-3b, however, shows the DHOST record as it would appear within the XNIM after being translated to Network Normal Form (NNF) by the Record Data Translator (RDT) portion of the Record Translator/Transformer (RTT) in step 6. Notice that trailing blanks have been dropped in CHARACTER data items, INTEGER and REAL data elements have been given explicit signs as well as decimal points, as appropriate, and BINARY fields have been "exploded" into character strings. In addition, the BOOLEAN field, DISABLED-VET, is now represented by the string '*T*'.

```
BKINEGKDGBLANMGODHBLIIAEACACBJKAMCHCDGBJEOGFPCMBHMJEGPDHBJEOGCABAA
IAEACABIIMAGEDGBMIMIGACAAPAPAPAPAPAPAPAPAPAIELCAAAAAAAAAAADCCGIIAE
ACADGJIENIGBDJBKEMCCABAAIAEACABAAIAEACABIAMEGADBBMINAGACADDJKENMFP
DAJLIMIFPDKBLMNMGJDBIIAEACADKBJENMGEDHJLINCHEDEJMMEACA
BAAIAEACABIAMIGADCBMINEGACADBJLMOEHEDEJMMNOGODCIIAEACABAAIAEACADJJ
NAOEGFDIAIAEACABAAIAEACABAAIAEACABIAMMGADDBMINIGACADIBJENMGJDBJKEN
IGMDEJLIEACABAAIAEACADDBLAOKCABAAIAEACABAAIAEACABAAIAEACABIANAGADE
BMINMGACADJBJEOGHEBAAIAEACABAAIAEACABAAIAEACACJJNEOGGB
DHBHMIEFPCAJLIOIGIDHJLIPCCABAAIAEACACBJKAMCHCDGBJEOGFPCFBLMNMGFDJI
IAEACABAAIAEACAAAAAAAABEAAAAAABEAAAAAAAAA
<<>>
```

## FIGURE4-3a: XNIM-BASED CHARACTER REPRESENTATION OF DHOST RECORD

```
555667777\Charles X Jones     \1026920\+000111100001111000001
111000011110000\-1110000111100001111000011110000 1111\+150.25
\+50\M\malaria     \0101940\gin and tonic   \tendonitis
  \0202950\cortisone     \strep     \0303960\penicillin
    \flu     \0404970\rest     \Susan B Anthony
    \Charles Jones     \+20\+20\*F*\
```

## FIGURE 4-3b: DHOST RECORD IN NNF

Once the record is in this form, it is ready for input to the record transformation routine (RTR) at step 7. RDT handles all translation to and from Network Normal Form, while RTR performs the required operations (e.g., +) on the data items, with the results shown in Figure 4-4a. This is actually the Network Normal Form of the record as expected by the PHOST. Note that all parent-related data has been dropped in this example, as no entry referring to those fields exists in the Transformation Description Table.

```
555667777\CHARLES X JONES   \M\40\1026920\ + 000000000000
0000000000000000000000\-111111111111111111111111111111\1
0.0\5\STREP  PENICILLIN  0303960\FLU  REST  0404970\*F*\
```

FIGURE 4-4a: PHOST RECORD IN NNF

Finally, this version of the record must be passed through the RDT at step 8 in order to produce the record in the native, PHOST format, required by PPROG. The resulting translated, transformed, and translated record, Figure 4-4b, is transmitted on to the awaiting PHOST process PPROG.

```
DFDFDGDFDHDGDHDHAADHGIEDHCGBGFGMFPHDFPFIGPEKGFGOCAHDCACACACAAAENAA
CIDADBDGDCDCDJAADAAAAAAAAAAAAAAAAEBKAAAAAHEHDGFHCCAHACACACACACACACA
HACAGOGFGDGJGMGJGJGMCAGOCACACACADDDADDDADGDJAADAGMGGCAHFCACACACACA
CACACACACAHCCAHDGFCAHECACACACACACACACACACADEDADEDADHDJAADAAAAA
```

FIGURE 4-4b: XNIM-BASED CHARACTER REPRESENTATION OF PHOST RECORD

In this example, PPROG then displays the record as shown in Figure 4-5.

```
(phost)Output is:
ssn[c9] = 555667777
name [c20] = Charles X Jones
sex[c1 = M
yrs g[i2] = 40
birth [c7] = 1026920
teste [b36] = 0
suspe[b36] = 0
wt[i3] = 0
ht[r3] = 5.0000000
dis1[c37] = strep  penicillin  0303960
dis2[c37] = flu  rest  0404970
```

FIGURE 4-5: DISPLAY OF DATA AT PHOST

## 5. PERFORMANCE CONSIDERATIONS

Once the feasibility of a concept has been demonstrated, questions naturally arise regarding the practicality of the approach. Practicality devolves into two issues: i) feasibility of implementation, and ii) performance of the result. Based upon our implementation of an RRA mechanism within the ICST Experimental Computing Facility, we have no reason to believe that the construction of a production mechanism for supporting of data transmission between heterogeneous systems poses any major problems. Thus, it is appropriate to consider the performance issue.

To estimate RRA bandwidth, we will assume that both request and response packets are approximately 1000 bits in length, that both request and response travel through two intermediate packet switches and that the average distance between packet switches is 500 miles. Using 100,000 miles per second as the speed of electric flow in copper wires, it follows that the average time for a packet to move between packet switches is 5 ms. Moreover, assuming 50 Kb. lines, the average time to encode a packet is 20 ms. A total of three encodings are required (source, and two intermediate nodes). Thus, the average time for a packet to travel from source to destination is 75 ms. excluding processing and queuing times. It follows that the round trip time is 150 ms. It follows that even if the remote data could be instantaneously transferred into a buffer, the maximum processing rate would be approximately 6.6Kb/second and the bandwidth against the DBMS would be approximately 6.6 Kb. Since accessing data in remote systems is likely to require a significant amount of time, the actual bandwidth is likely to be significantly lower, perhaps on the order of 1-2 Kb.

The preceding result is of more than passing interest. To provide an appropriate context, we observe in accordance with Scott-Morton [LUCAH 75] that information processing can be divided into three major categories: operational control, managerial control, and strategic planning. As one passes from operational control to strategic planning both the bandwidth and the predictability of the requirement decrease. Thus, operational control applications are typically high bandwidth and very predictable, e.g. payroll. In contrast, strategic planning requirements are intrinsically low bandwidth and very unpredictable, e.g. which ships are close to a country undergoing a revolution.

Given this context, we are led to conclude that remote access to data in support of operational control is likely to be unsatisfactory. In support of managerial control, it may be unsatisfactory, and in support of strategic planning it is likely to be very satisfactory. As a close corollary, a generalized principle of locality applies. This principle states that: "remote data should be rarely accessed."

In considering these somewhat philosophical comments, it is important to bear in mind that they are predicated on existing communications technology, e.g. relatively low bandwidth, relatively high cost communications based on using circuits provided by common carriers. Satellite transmission promises a much higher bandwidth at a much lower cost. Nevertheless, in view of transmission delays (.5 seconds round trip), it is still unlikely that high bandwidth remote applications can be effectively supported unless there is a very substantial predictability in the data to be accessed. That, is applications in which large amounts of data can be prespecified are likely to prove more appropriate than those for which it is infeasible to predict future data requirements.

## 6. STRUCTURED DATA TRANSFER PROTOCOLS

If the requirements for a RRA capability are examined from a more general perspective, insight is gained regarding the specification of basic protocols supporting the exchange of structured data.

A Structured Data Transfer Protocol (SDTP) may be viewed as a mechanism which facilitates the sharing of structured data between processes in a computer networking environment. Such exchange of structured data between processes mandates a means of specifying and executing a transformation between different physical and organizational data formats and representations. Specification, creation and/or selection of records to be exchanged via a SDTP would be included in the set of responsibilities of the processes invoking the SDTP.

A specification for a SDTP would consist of the following:

1. a standard format for the exchange of structured data.

2. the information required to describe the exchanged data.

3. the control information (i.e., commands) needed to signal the establishment, maintainence, operation and termination of a connection between SDTP processes.

4. flow and error control responsibilities.

A SDTP would assume the existance of lower level services which would provide the means for reliable transport of information between specified, cooperating processes on a network. It should support interactive use by both humans and processes. Consequently, its operation much be completely deterministic.

Existing standards could prove useful in the development of a standard SDTP. Among these are the ANSI Code for Information Interchange (ASCII) [ANSI 1,2], the standard for character representation of numeric values [ANSI 3], the standard format for exchanging bibliographic information on magnetic tape [ANSI 4], and the proposed standard for data descriptive files [ANSI 5].

In conjunction with the selection of standard formats, an assessment could be made of current and projected requirements for structured data interchange. For example, the cost benefits of providing SDTP support for only character-encoded, structured data should be considered, vs. those for full support of other data types (e.g., binary, real). In addition, the cost vs. benefits of developing and using a SDTP supporting self-describing data (e.g., [ANSI 5]) vs. the transmission of data independently of descriptive information (e.g., XRRA approach) should be evaluated.

## 7. DIRECTIONS OF FUTURE WORK

Remote Record Access is a prime component of general purpose network operating systems. The design and implementation of the described capability has provided a wealth of information about the capabilities and limitations of various approaches to exchanging structured data. This knowledge in turn can prove useful in the development of specifications for (much needed) Structured Data Transfer Protocols. Higher level data sharing services, such as those supporting structured file transfer and distributed data base management, may in turn be built upon such a foundation.

Widespread interest in and use of data base management systems has stimulated investigations into the implications of marrying computer networking and DBMS technology [BOOTG 72,76] [BERGJ 76] [KIMBS 79]. The rapidly growing dependence on computer networking technology to meet information management and communications needs in government and industry, suggests that the time is "ripe" for development of standardized, high level communications protocols including those for structured data transfer.

Efforts are underway nationally and internationally to develop standards which will facilitate the use of computer networking technology (e.g., ANSI, ISO, CCITT). At the National Bureau of Standards, the development of high level computer networking protocols is part of a larger effort geared towards the development of an entire "family" of computer system and network standards. These are intended to permit the successful interconnection of competitively procured computer system and network components. Through the development and use of such standards it is believed that the performance and cost advantages of competitively procured systems and components can be used to full advantage by Federal agencies, while at the same time assuring reliable and efficient system operation.

## REFERENCES


[ANDEA 71] Anderson, A., et. al., The Data Reconfiguration Service - An Experiment in Adaptable Process/Process Communication, Proceedings of the Second Symposium on the Optimization of Data Communication Systems, IEEE Press, October 1971.

[ANDEA 74] Anderson, A.K., J.W. Benoit, M.A. Padlipsky, Description of the Prototype WWMCCS Intercomputer Network (PWIN) Protocols, Mitre Corporation, MTR-5444, December 1974.

[ANSI 1] Code Extension Techniques for Use with the 7-Bit Coded Character Set for American National Standard Code for Information Interchange (FIPS 35), American National Standards Institute, X3.41-1974.

[ANSI 2] Code for Information Interchange, American National Standards Institute, X3.4-1977.

[ANSI 3] Representation of Numeric Values in Character Strings for Information Interchange, American National Standards Institute, X3.42-1975.

[ANSI 4] Standard for Bibliographic Information Interchange on Magnetic Tape, American National Standards Institute, Z39.2-1971.

[ANSI 5] American National Standard Specification for an Information Interchange Data Descriptive File, American National Standards Institute, X3L5-1978.

[BACHM 79] Bach, M. J., N. H. Goguen, M. M. Kaplan, "The Adapt Data Translation System and Applications," Proceedings of the Fourth Berkeley Conference on Distributed Data Management and Computer Networks, August 1979.

[BERGJ 76] Berg, John L. (ed.), Data Base Directions: The Next Steps, (proceedings of the Workshop of the National Bureau of Standards and Association for Computing Machinery, October, 1975), National Bureau of Standards, Special Publication 451, September 1976.

[BIRSE 76] Birss, Edward W., and James P. Fry, "Generalized Software for Translating Data," Proceedings of 1976 National Computer Conference, AFIPS Press, Montvale, New Jersey, 1976, pp. 889-897.

[BOOTG 72] Booth, Grayce M., "The Use of Distributed Data Bases in Information Networks," Computer Communication: Impacts and Implications, First International Conference on Computer Communications (October 1972), pp. 371-376.

[BOOTG 76] Booth, Grayce M., "Distributed Information Systems," Proceedings 1976 National Computer Conference, AFIPS Press, Vol. 45, pp. 789-794.

[CERFV 72] Cerf, Vinton G., Eric Harslem, John Heafner, Robert Metcalfe, and James White, "An Experimental Service for Adaptable Data Reconfiguration," IEEE Transactions on Communications, 20:3, (June 1972), pp. 557-564.

[CROCS 72] Crocker, S.D., J.H. Heafner, R.M. Metcalfe, and J.B. Postel, "Function Oriented Protocols for the ARPA Network," Proceedings of the Spring Joint Computer Conference, AFIPS Press, Vol. 40, 1972, pp. 271-279.

[FITZM 78] Fitzgerald, M.L., Common Command Language for File Manipulation and Network Job Execution: An Example, National Bureau of Standards, Special Publication 500-37, August 1978.

[FOLTH 78] Folts, H.C., "Evolution Toward a Universal Interface for Data Communications," Proceedings of International Conference on Computer Communications, 1978.

[FORSH 77] Forsdick, H.C., R.E. Schantz, and R.H. Thomas, "Operating Systems for Computer Networks," BBN Report No. 3614, Bolt Beranek and Newman, Cambridge, Mass., 1977.

[FRYJ 72A] Fry, James P., Diane P. Smith, and Robert W. Taylor, "An Approach to Stored Data Definition and Translation," Stored Data Definition and Translation (SDDT) Task Group, Proceedings of 1972 ACM SIGFIDET Workshop: Data Description, Access and Control, A.L. Dean (ed.), Association for Computinq Machinery, 1972

[FRYJ 72B] Fry, James P., Randall L. Frank, and Ernest A. Hershey III, "A Developmental Model for Data Translation," Proceedings of 1972 ACM SIGFIDET Workshop: Data Description, Access and Control, A.L. Dean (ed.), Association for Computing Machinery, 1972.

[FRYJ 74] Fry, James P., and David W. Jeris, "Towards a Formulation and Definition of Data Reorganization," Proceedings of 1974 ACM SIGMOD Workshop on Data Description, Access and Control, Randall Rustin (ed.), Association for Computing Machinery, 1974, pp. 83-100.

[FRYJ 78] Fry, James, Internal report prepared for National Bureau of Standards, 1978.

[HARSE 71] E.F. Harslem and J.F. Heafner, The Data Reconfiguration Service - An Experiment in Adaptable, Process/Process Communication, RAND Corporation, R-860-ARPA, November 1971.

[HARSE 72] Harslem, E., and J. Heafner, The Data Reconfiguration Service Compiler: Communication Among Heterogeneous Computer Centers Using Remote Resource Sharing, RAND Corporation Technical Report R-887-ARPA, April, 1972.

[HOUSB 77] Housel, B.C., N.C. Shu, R.W. Taylor, S.P. Ghosh, and V.Y. Lum, EXPRESS: A Data Extraction, Processing, and Restructuring System, IBM Research Laboratory, RJ1962(27742), 1977.

[ISO 79] ISO/TC97/SC16 N227, "Reference Model of Open Systems Interconnection," August 1979.

[KIMBS 76] Kimbleton, S.R. and R.L. Mandell, "A Perspective on Network Operating Systems," Proceedings 1976 National Computer Conference, AFIPS Press, Montvale, N.J., Vol. 45, 1976, pp. 551-559.

[KIMBS 78] Kimbleton, Stephen R., Helen M. Wood, and M.L. Fitzgerald, "Network Operating Systems -- An Implementation Approach," Proceedings 1978 National Computer Conference, AFIPS Press, Montvale, N.J. Vol. 47, 1978, pp. 773-782.

[KIMBS 79] Kimbleton, Stephen R., Pearl S.-C. Wang, and Elizabeth N. Fong, "XNDM: An Experimental Network Data Manager," Proceedings of the Third Berkely Workshop on Distributed Data Processing, 1979.

[LEVIP 77] Levine, Paul H., Facilitating Interprocess Communication in a Heterogeneous Network Environment, MIT/Laboratory for Computer Science, MIT/LCS/TR-184, July 1977.

[LUCAH 75] Lucas, H. C. Jr., "Why Information Systems Fail," Columbia University Press, New York, 1975.

[MARIT 75] Marill, Thomas, and Dale Stern, "The Datacomputer - A Network Data Utility," Proceedings of the National Computer Conference, 1975, AFIPS Press, pp. 389-395.

RTA 74] Merten, Alan G., and James P. Fry, "A Data Description Language Approach to File Translation," Proceedings of 1974 ACM SIGMOD Workshop on Data Description, Access and Control, Randall Rustin (ed.), Association for Computing Machinery, 1974, pp. 191-205.

THJ 77] Rothnie, J.B., and N. Goodman, An Overview of the Preliminary Design of SDD-1: A System for Distributed Databases, Computer Corporation of America, Technical Report CCA-77-04, March 1977.

HNG 75A] Schneider, G. Michael, "DSCL - A Data Specification and Conversion Language for Networks," Proceedings of ACM SIGMOD Conference, May 1975, Association for Computing Machinery, 1975, pp. 139-148.

HNG 75B] Schneider, G. Michael, and E.J. Desautels, "Creation of a File Translation Language for Networks," Information Systems, 1:1, (January 1975), pp. 23-31.

OSA 72] Shoshani, Arie, "Data Sharing in Computer Networks," Proceedins of 1972 Wescon Electronic Show and Convention, September 1972, Institute of Electrical and Electronics Engineers.

IOSA 73] Shoshani, A., and I. Spiegler, "The Integration of Data Management Systems on a Computer Network," Proceedings of AIAA Computer Network Systems Conference, American Institute of Aeronautics and Astronautics, 1973, AIAA Paper No. 73-417.

IUN 76] Shu, N.C., V.Y. Lum, B.C. Housel, An Approach to Data Migration in Computer Networks. IBM Research Laboratory, 1976.

IUN 77] Shu, N. C., B. C., Housel, R. W. Taylor, S. P. Ghosh and V. Y. Lum, "EXPRESS: A Data EXtraction, Processing, and REstructuring System," ACM Transactions on Database Systems, Vol. 2, No. 2, June 1977.

MPR 78] Simpson, R. O., G. H. Phillips, "Network Job Entry Facility for JES2", IBM Systems Journal, International Business Machines Corporation, Vol. 17, No. 3, 1978.

PPC 72] Sippl, Charles J., and Charles P. Sippl, Computer Dictionary and Handbook, Howard W. Sams and Co., Inc., 1972.

MITD 72] Smith, Diane C.P., "A Method for Data Translation Using the Stored-Data Definition and Translation Task Group Languages," Proceedings of 1972 ACM SIGFIDET Workshop: Data Description, Access and Control, A.L. Dean (ed.), Association for Computing Machinery, 1972.

[TREMJ 76] Tremblay, J. P., An Introduction to Data Structures with Applications, McGraw-Hill, Inc. 1976.

NBS-114A (REV. 2-80)

| U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions) | 1. PUBLICATION OR REPORT NO. NBS SP 500-71 | 2. Performing Organ. Report No. | 3. Publication Date December 1980 |
|---|---|---|---|

**4. TITLE AND SUBTITLE**

Computer Science & Technology:
Remote Record Access: Requirements, Implementation and Analysis

**5. AUTHOR(S)**

Helen M. Wood and Stephen R. Kimbleton

**6. PERFORMING ORGANIZATION** (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

**7. Contract/Grant No.**

**8. Type of Report & Period Covered**

Final

**9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS** (Street, City, State, ZIP)

Same as above.

**10. SUPPLEMENTARY NOTES**

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

**11. ABSTRACT** (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

A key support component for network-wide data sharing is the ability of a process to access remotely stored data at runtime. In order for the accessed data to be useful, a means of overcoming differences in data representation and format is necessary. Such a capability is termed remote record access. This paper identifies some of the problems inherent in the sharing of data among dissimilar computer and data systems. Implementation issues and alternatives are presented, followed by a description of XRRA, the Experimental Remote Record Access component which has been implemented as part of the Experimental Network Operating System (XNOS) at the National Bureau of Standards.

**12. KEY WORDS** (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Computer networking; data conversion; data transformation; data transfer; data translation; network operating systems.

**13. AVAILABILITY**

[X] Unlimited
☐ For Official Distribution. Do Not Release to NTIS
[X] Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

**14. NO. OF PRINTED PAGES**

41

**15. Price**

$2.25

# CHECK THEM OUT.

"AUTOMATION IN THE MARKETPLACE"
A Consumer's Guide

D/H CAKE MIX .83
WALNUTS CAN .59
JELLO PUDDING .30

▶ UNIVERSAL PRODUCT CODE see booklet

1 GREEN PEPPER .34
LASER SCANNER see bklt
CHRRY TOMATO .79

▶ ELECTRONIC CASH REGISTER see bklt

1 CUCUMBERS .34

▶ HANDLING OF UNCODED ITEMS see bklt

▶ ELECTRONIC SCALES see bklt

GREETING CARD .60

▶ WEIGHTS & MEASURES ENFORCEM'T see bklt

DELICATESSEN 1.35
2.19b @49/bBROCCO 1.07

▶ SPECIAL FEATURES OF COMPUTER CHECKOUT SYSTEMS see bklt

DRUG 4.49 T

▶ BANK TELLER MACHINES see bklt

▶ COMPUTER TERMINALS see bklt

▶ CONSUMER ISSUES see bklt

▶ THANK YOU BE INFORMED see bklt

łow do those automated checkout counters, gas pumps, ;redit offices, and banking facilities work? What every ;onsumer should know about the modern electronic sys- ;ems now used in everyday transactions is explained in ı 12-page booklet published by the Commerce Depart- nent's National Bureau of Standards.

**\utomation in the Marketplace** (NBS Consumer Informa- ion Series No. 10) is for sale by the Superintendent of )ocuments, U.S. Government Printing Office, Washington, ).C. 20402. Price: 90 cents. Stock No. 003-003-01969-1.

**(please detach along dotted line)**

---

## ORDER FORM

PLEASE SEND ME ____ COPIES OF

**Automation in the Marketplace**

ıt $.90 per copy.

ɨ.D. Stock No. 003-003-01969-1

I enclose $____________ (check, or money order) or charge my Deposit Account No.____________________.

Total amount $____________________.

Make check or money order payable to Superintendent of Documents.

**MAIL ORDER FORM WITH PAYMENT TO**

**Superintendent of Documents**
**U.S. Government Printing Office**
**Washington, D.C. 20402**

or any U.S. Department of Commerce district office

(please type or print)

NAME________________________________________

ADDRESS______________________________________

CITY_________________________________________

STATE_____________________ ZIP CODE______________

**FOR USE OF SUPT. DOCS.**

____Enclosed ________
To be mailed
____later ________
Refund ________
Coupon refund ________
Postage ________

☆ U.S. GOVERNMENT PRINTING OFFICE 1980-340-997/302

# NBS TECHNICAL PUBLICATIONS

## PERIODICALS

**JOURNAL OF RESEARCH**—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $13; foreign $16.25. Single copy, $3 domestic; $3.75 foreign.

NOTE: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

**DIMENSIONS/NBS**—This monthly magazine is published to inform scientists, engineers, business and industry leaders, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing. Annual subscription: domestic $11; foreign $13.75.

## NONPERIODICALS

**Monographs**—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

**Handbooks**—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

**Special Publications**—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

**Applied Mathematics Series**—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

**National Standard Reference Data Series**—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

**Building Science Series**—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

**Technical Notes**—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

**Voluntary Product Standards**—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

**Consumer Information Series**—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

*Order the* **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

*Order the* **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Services, Springfield, VA 22161.*

**Federal Information Processing Standards Publications (FIPS PUB)**—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

**NBS Interagency Reports (NBSIR)**—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services, Springfield, VA 22161, in paper copy or microfiche form.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

SPECIAL FOURTH-CLASS RATE
BOOK