NTIA-REPORT-78-4

Digital Communication Performance Parameters for Proposed Federal Standard 1033

VOLUME I: STANDARD PARAMETERS

N. B. Seitz
P. M. McManamon

U.S. DEPARTMENT OF COMMERCE
Juanita M. Kreps, Secretary

Henry Geller, Assistant Secretary-Designate
for Communications and Information

May 1978

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DIGITAL COMMUNICATION PERFORMANCE PARAMETERS FOR PROPOSED FEDERAL STANDARD 1033

VOLUME I: STANDARD PARAMETERS

Neal B. Seitz and Peter M. McManamon*

There is a growing need within the Federal Government for uniform means of specifying and measuring the performance of data communication systems from the point of view of the digital services delivered to the end user. This report presents results of a cooperative interagency program aimed at meeting this need through the development of universally applicable, user-oriented performance parameters. The defined parameters have been incorporated in a proposed Federal Standard which is currently undergoing review and coordination within government and industry.

1. INTRODUCTION

1.1. Background

In March of 1973, the Federal Telecommunications Standards Committee (FTSC) assigned the Institute for Telecommunication Sciences (ITS) responsibility for FTSC Technical Subcommittee 2, charged with developing a new Federal Telecommunications Standard: "Digital Communication Performance Parameters." The purpose of the new standard (proposed Federal Standard 1033) was to specify a set of parameters which could be used to describe the performance of any data communication system from the point of view of the digital services[1] delivered to the end user. Parameters selected for inclusion in the standard were to have the following general attributes:

*The authors are with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80303.

[1]Except as noted, the term "services" always implies digital communication services.

1. <u>User Orientation</u>. Selected parameters would describe the performance of services delivered to the end user, rather than the performance of equipment or facilities used to provide such services. The parameters would describe performance in terms of events directly observable to end users. The parameters would be chosen on the basis of performance concerns expressed by end users, rather than on the basis of engineering design considerations.
2. <u>Universal Applicability</u>. Selected parameters would not be restricted, in definition or application, to particular telecommunication systems or classes of systems. The parameter definitions would be independent of network-unique characteristics such as topology and control protocol.
3. <u>Simplicity</u>. Selected parameters would be simple enough to be readily understood by non-technical users; and would be defined such that users could obtain accurate measured values, within reasonable measurement intervals, using simple, inexpensive test equipment such as counters and timers. Wherever possible, parameters would be measurable during normal operational use of a service, without the need for special test scenarios.
4. <u>Completeness</u>. The set of selected parameters would encompass, and provide some relevant information on, all performance factors of significance to telecommunications users. The parameters would be "well-behaved" in the sense that they would reliably reflect actual performance over the full range of possible parameter values.

It was anticipated that a set of parameters with these attributes would be useful to Federal agencies in specifying performance requirements for end-to-end services; allocating end-to-end requirements to purchasable items; evaluating equipment and service alternatives on a cost/performance basis; and monitoring delivered performance to ensure compliance with specified performance goals.

As it is currently written, Federal Standard 1033 will impose two principal requirements on Federal agencies responsible for supplying digital telecommunication services. The first requirement stipulates that such agencies must specify the end-to-end performance of services they supply, prior to procurement, by providing numerical values for all performance parameters defined in the standard. Performance parameter values may be developed in either of two ways: (1) by analysis of the operational and functional requirements for a proposed system or service; or (2) by means of actual system performance measurements. The second requirement defines the user/telecommunication system interface at which the specified performance parameter values are to apply. The interface is defined in functional terms, in such a way that communication processing functions such as error control, flow control, and polling are included within "the telecommunication system" irrespective of the physical facilities in which they are performed.

In the course of its development, the scope of the proposed Federal Standard was restricted in a number of important ways. It was decided that the standard should not specify, or require that suppliers attain, particular numerical values for the defined performance parameters. Standard measurement methods were identified as the subject of a separate Federal Standard which is currently under development. It was decided that neither standard will require that Federal agencies conduct performance measurements at specified intervals; each agency will decide whether and when to measure performance based on its own needs. Since the standard applies only to Federal agencies, it imposes no requirement on common carriers or other non-Federal organizations. The standard will present carriers with an opportunity to specify performance in terms of the defined parameters, however, since services procured by Federal agencies will be described in these terms. The standard places no restriction on the use of other performance parameters in Federal procurement specifications, as long as such additional parameters are not used in lieu of the standard parameters.

Technical Subcommittee 2 submitted proposed Federal Standard 1033 to the full FTSC committee at the end of Fiscal 1976. After a preliminary review and evaluation, the FTSC approved preliminary publication of the proposed standard in the Federal Register to solicit public comment (National Communications System, 1977). This step is frequently undertaken to achieve a wider consensus for proposed Federal Standards prior to their approval.

Comments on the proposed standard were received from eleven separate organizations, including most of the major national and international common carriers and several major suppliers of data processing equipment. These comments are being reviewed for incorporation in a revised version of the proposed standard. It is anticipated that FTSC will submit this revised standard to its member agencies for formal coordination and approval.

1.2. Purpose and Scope of Report

The purpose of this two-volume NTIA Report is to assist Federal agencies and other organizations in reviewing proposed Federal Standard 1033, by describing its underlying technical foundation and illustrating its intended application. Volume I provides a detailed description of the technical considerations which influenced development of the standard; and defines the selected parameters in logical and mathematical terms. Volume II illustrates application of the standard in specifying the performance of three example telecommunication services. Both volumes are intended to supplement, rather than to supplant, the standard. In any case of conflict between the information in these reports and that in the standard, the latter should be considered superseding.

Figure 1.1 illustrates NTIA's Data Communications project, which served as a focal point for development of the standard. The overall project is comprised of three major project activities:

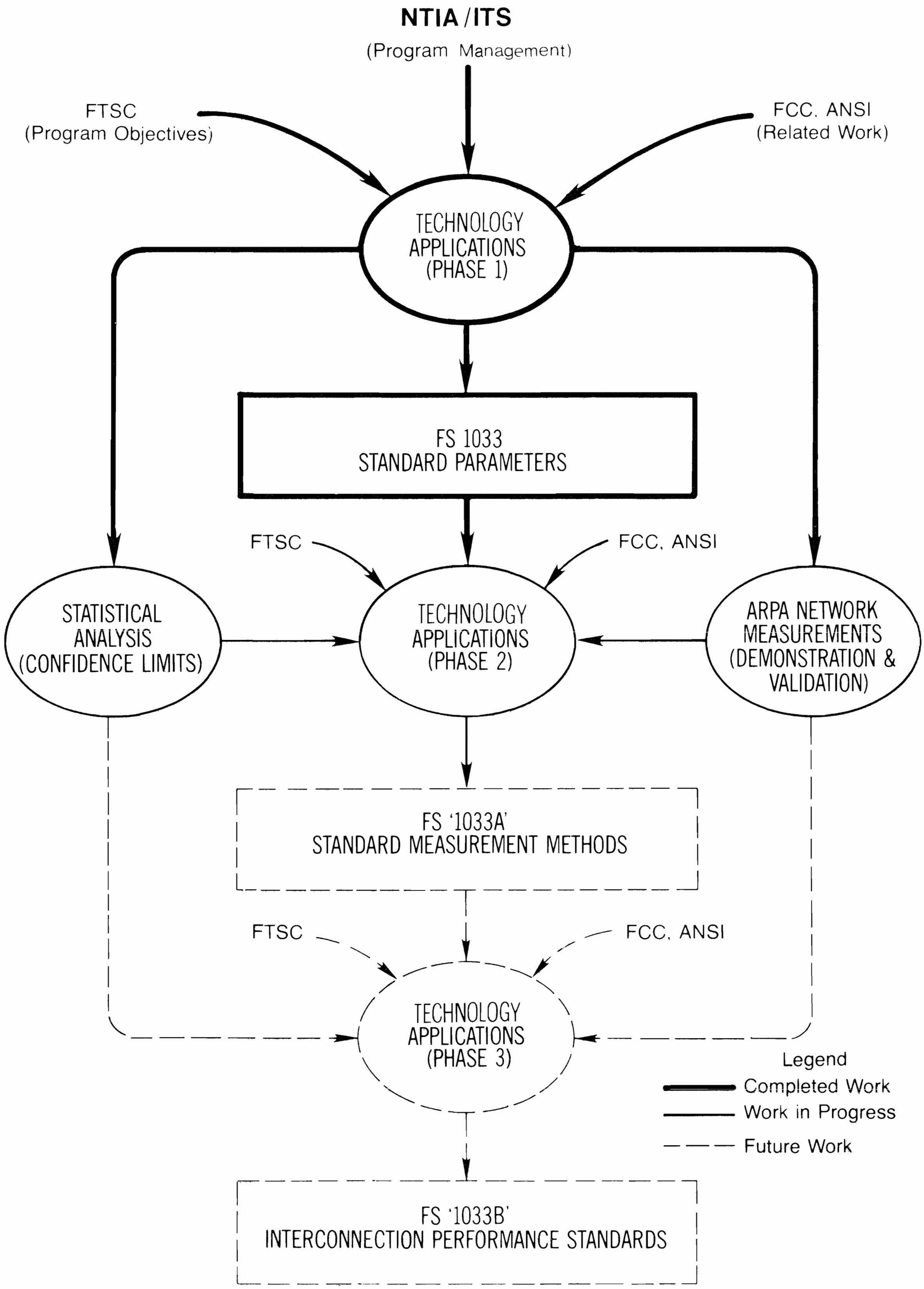


Figure 1.1. Data communications project overview.

Technology Applications (consisting of three successive phases); Statistical Analysis; and ARPA Network Measurements. Proposed Federal Standard 1033 is primarily an output of the Technology Applications activity, Phase I. Interim results of this activity have also been presented in two earlier reports (McManamon et al., 1975; Seitz and McManamon, 1976a) and three published articles (Seitz and McManamon, 1976b; Seitz and McManamon, 1977a; Seitz and McManamon, 1977b).

The second major Data Communications project activity, Statistical Analysis, is aimed at developing confidence limits and sample size requirements to standardize measurement of the selected performance parameters. These statistical results are essential to the planned measurement standard. Without such data, there is no basis for agreement between the suppliers and users of a service as to the validity of performance measurements. Results of this activity have been presented in two reports (Crow, 1974; Crow and Miles, 1977) and a number of published articles (e.g., Crow, 1975; Crow and Miles, 1976).

The third major Data Communications project activity, ARPA Network Measurements, has been undertaken to substantiate the theoretical results of the first two activities by means of actual network performance measurements. Initial results of this activity have been presented in an NTIA Report (Payne, 1978).

1.3. Parameter Development Process

Figure 1.2a summarizes the process by which the proposed standard parameters were developed. The process comprised four major steps:

1. System Analysis. Existing and proposed telecommunication services were surveyed to characterize the user/system interface signals exchanged during an end-to-end information transfer transaction. Results of the analysis were consolidated in a simple functional model which is believed to be capable of representing any digital telecommunication process from the end user point of view.

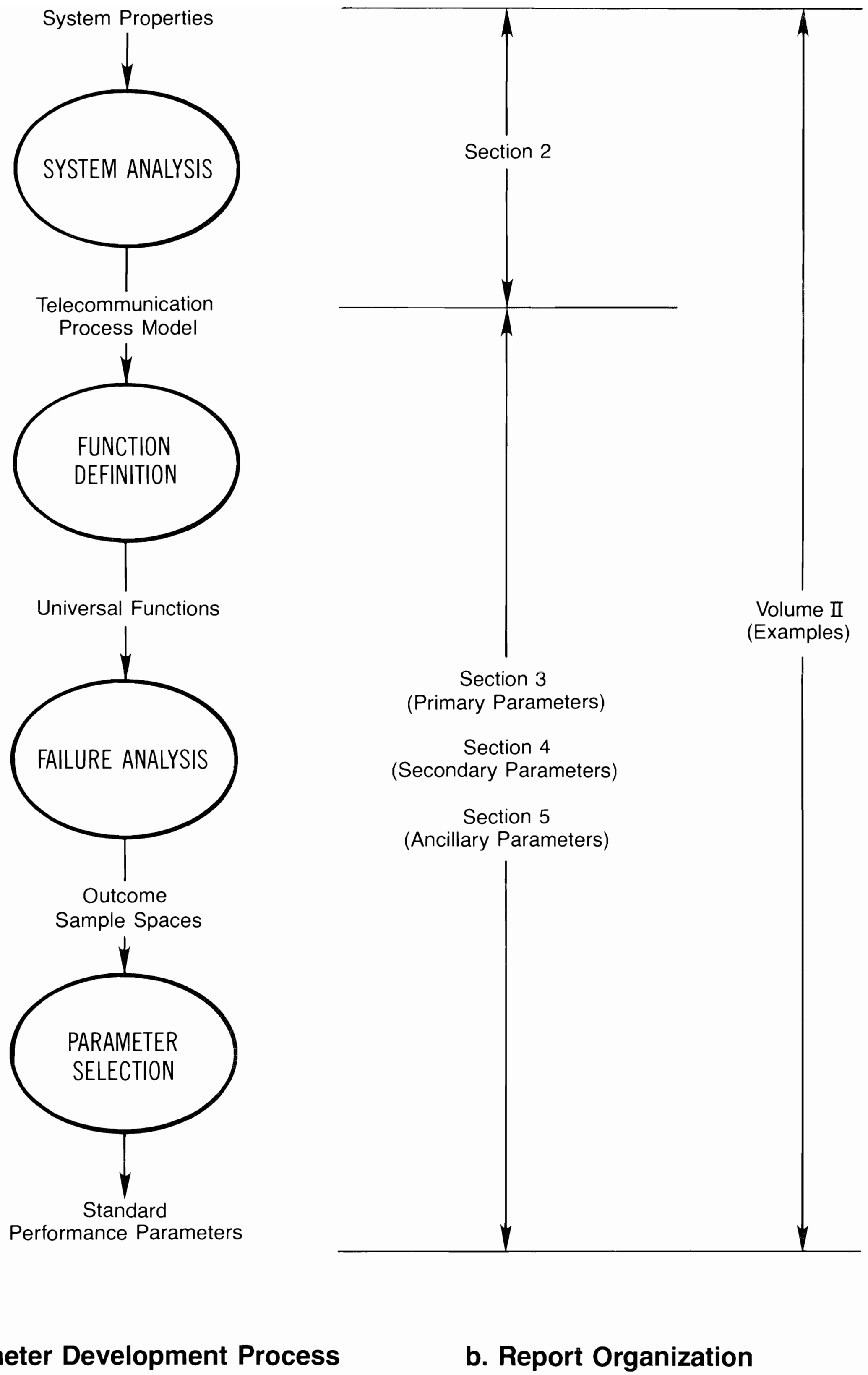


Figure 1.2. Plan of the work.

2. Function Definition. The telecommunication process model was used as the basis for defining a set of "universal" communication functions. The defined functions are universal in the sense that they necessarily form a part of any useful telecommunication process.
3. Failure Analysis. Each of the defined communication functions was analyzed to determine the possible outcomes an individual "trial performance" might encounter. For each function, the possible outcomes judged to be of separate concern to the end user were represented in a "pie" diagram of outcome possibilities called a "sample space."
4. Parameter Selection. Candidate performance parameters were derived from the sample spaces. Parameters were evaluated on the basis of user orientation, simplicity, and completeness.

1.4. Report Organization

Figure 1.2b summarizes the overall organization of the report as it relates to the four-step parameter development process described above. The remainder of this volume is divided into four sections:

Section 2 provides a brief review of fundamental modeling concepts; defines the telecommunication process model; and illustrates the general operation of the model in terms of two familiar user/system interaction sequences.

Section 3 defines a set of "primary parameters" which express system performance relative to five primary functions: access, bit transfer, block transfer, message transfer, and disengagement.

Section 4 defines a set of "secondary parameters" which express system performance relative to two secondary functions: service continuation and service restoral.

Section 5 defines a set of "ancillary parameters" which express, for each primary function, the average proportion of total performance time attributable to user performance delay.

Sections 3, 4, and 5 each develop their respective parameters in three distinct steps, as indicated in Figure 1.2b: Function Definition, Failure Analysis, and Parameter Selection. Volume II of the report illustrates the application of all four parameter development steps to the specification of performance for particular systems.

The distinction between the primary, secondary, and ancillary measures will become clear as the parameters are developed in their associated sections. Briefly, the primary functions and outcomes are defined directly on the telecommunication process model, in terms of particular model state transitions; and the associated parameters are intended to provide a detailed description of performance by focusing on a relatively short observation period. As an example, the parameter Access Time describes performance expected during the access phase of an individual information transfer transaction.

The secondary functions and outcomes are defined on the basis of primary parameter values, rather than directly on the model; and the associated parameters are intended to provide a more macroscopic, longer-term view of performance. As an example, the performance parameter Outage Duration expresses the average duration of system outages observed by the user. This parameter is necessarily more general than the primary parameters, since outages can only be defined in terms of measured primary parameter values. The secondary parameters are closely associated with the traditional concept of availability.

The purpose of defining ancillary outcomes and parameters for the primary functions is to provide a quantitative means of expressing the influence of user performance on overall performance of these functions. The ancillary outcomes and parameters are defined on the basis of model "responsibility state" transitions occurring during performance of an associated primary function. As an example, in the public switched network, the period between off-hook and dial tone represents an interval of "system responsibility", and the period between ringing and answer represents an interval of "user responsibility", during the primary function of access. The ancillary parameter User Access Time Fraction describes the average proportion of total Access Time during which the user is responsible for advancing the access attempt to successful completion.

Figure 1.3 summarizes all performance parameters selected for inclusion in the proposed Federal Standard. A total of 26 parameters are specified, including 19 primary parameters; 3 secondary parameters; and 4 ancillary parameters. Each selected parameter is defined in two ways:

1. Axiomatically, by reference to the "pie" diagram or sample space for the corresponding communication function.
2. Mathematically, by reference to parameter definition flowcharts and associated equations.

The parameter definition flowcharts and equations provide a procedure for calculating values for the defined parameters, based on a measured sample population of performance trials.

2. TELECOMMUNICATION PROCESS MODEL

2.1. Introduction

Any description of performance ultimately refers to some functional model. This is clearly true in the case of performance specification, since the performing entity normally is not selected (and may not even exist) until after the specification is developed. But even in evaluating the performance of an existing system, we never look at everything the system does; rather, we observe some subset of the system's activities, and compare these observations with an abstract idea of what the

FUNCTION	PERFORMANCE CRITERION			
	EFFICIENCY	ACCURACY	RELIABILITY	
ACCESS (a)	• ACCESS TIME $W(a_s)$	• INCORRECT ACCESS PROBABILITY $P(a_m)$	• ACCESS DENIAL PROBABILITY $P(a_\ell)$	• USER ACCESS TIME FRACTION p(a)
BIT TRANSFER (b1)	• BIT TRANSFER TIME $W(b1_s)$	• BIT MISDELIVERY PROBABILITY $P(b1_m)$ • BIT ERROR PROBABILITY $P(b1_e)$ • EXTRA BIT PROBABILITY $P(b1_x)$	• BIT LOSS PROBABILITY $P(b1_\ell)$	• USER BLOCK TRANSFER TIME FRACTION p(b2)
BLOCK TRANSFER (b2)	• BLOCK TRANSFER TIME $W(b2_s)$	• BLOCK MISDELIVERY PROBABILITY $P(b2_m)$ • BLOCK ERROR PROBABILITY $P(b2_e)$ • EXTRA BLOCK PROBABILITY $P(b2_x)$	• BLOCK LOSS PROBABILITY $P(b2_\ell)$	
MESSAGE TRANSFER (b3)	• BIT TRANSFER RATE $R(b1_s)$ • BIT RATE EFFICIENCY $Q(b1_s)$ • BLOCK TRANSFER RATE $R(b2_s)$ • BLOCK RATE EFFICIENCY $Q(b2_s)$	• OUTAGE PROBABILITY $P(b3_z)$		• USER MESSAGE TRANSFER TIME FRACTION p(b3)
DISENGAGEMENT (d)	• DISENGAGEMENT TIME $W(d_s)$	• DISENGAGEMENT DENIAL PROBABILITY $P(d_\ell)$		• USER DISENGAGEMENT TIME FRACTION p(d)
SERVICE CONTINUATION (y)	• SERVICE TIME BETWEEN OUTAGES W(y*)			
SERVICE RESTORAL (z)	• OUTAGE DURATION W(z*)			

Legend

- Primary Parameters (unshaded)
- Secondary Parameters (cross-hatched)
- Ancillary Parameters (hatched)

Figure 1.3. Summary of selected performance parameters.

system ideally "should do." The selected subset of activities and the associated performance ideal are, in fact, a model of the real system.

This section describes a discrete functional model of the digital telecommunication process which served as the basis for defining the primary performance parameters. The overall structure of the model was determined by the desired parameter attributes listed in Section 1.1: user orientation, universal applicability, simplicity, and completeness. The model is believed to be capable of representing any digital telecommunication process from the end user point of view, regardless of system design characteristics such as topology and control protocol. Although the primary application of the model is in representing end-to-end services, it can also be used to represent the operation of functional or physical entities within a telecommunication system, as long as their inputs and outputs consist of discrete signals. Application of the model in the latter context would be useful in allocating end-to-end performance requirements to purchasable components or services.

2.2. Fundamental Modeling Concepts

This subsection provides a brief overview of fundamental modeling concepts which are used in defining the telecommunication process model. The subsection is tutorial in nature and can be omitted by readers familiar with discrete system models, finite state machines, and the use of mathematical functions in describing system input/output behavior.

2.2.1. Properties of Discrete System Models

Discrete system models are formally defined in terms of six basic concepts: entity, attribute, state, activity, event, and state history. Briefly, an entity is any distinct object of interest in a system. Entities have defined properties of interest called attributes. An attribute may characterize a single entity or may express the relationship between two or more entities. The state of a model is a description of all the entities and attributes as they exist at a particular point in time. The overall state of a model is often represented by a binary state variable. The concepts of entity, attribute, and state all refer to the static properties of a system model.

The temporal or dynamic properties of a system model are expressed in the concepts of activity, event, and state history. An activity is any process that causes changes in model state; an event is a change in model state. Events thus define and delimit the activities of a model; and conversely, the activities comprise periods of time between events. Each model event occurs at a distinct event time, and no two events occur at exactly the same time. A state history is a chronological record of model states over a sequence of consecutive events.

These concepts are frequently illustrated using the familiar "supermarket model." In this model, the entities consist of the shoppers, shopping baskets, and checkout stations. These entities have attributes such as total number and availability; and interact in various relationships as the shoppers (e.g.) select baskets and enter checkout queues. Activities include shopping, queueing at the checkout stations, and checkout. The arrival of shoppers and their completion of shopping and checkout are examples of discrete events. The state of the supermarket model is a complete description of the above entities and attributes as they exist at a given point in the model's operation. An event-by-event record of successive model states comprises a state history.

Although no standard procedure for developing a discrete system model can be stated, analysts normally divide the process into four general steps:

1. Defining the purpose of the model, based on its ultimate use.
2. Establishing boundaries which separate the modelled system from its environment.
3. Defining entities and their attributes.

4. Specifying the interactions among model entities, and between the model entities and the external environment.

Since the purpose of a model determines its structure, it is obvious that there can be no one "right model" of a given system. The primary criteria for model optimality, given a system and a modelling objective, are (1) minimum complexity, and (2) maximum accuracy in representing the actual operation of the modelled process.

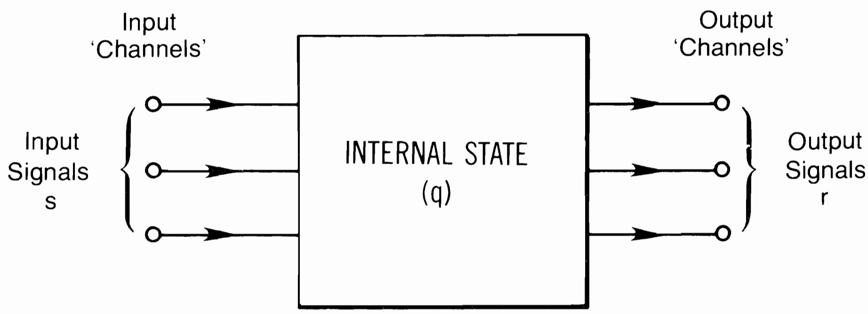
2.2.2. Finite State Machine Definition and Generalization

One very useful way of representing the entities in a discrete system model is the finite state machine (Fig. 2.1). A finite state machine can be viewed as a "black box" which interacts with its environment ("the user") via a finite set of input and output channels, each capable of communicating only a finite number of distinguishable signals (Minsky, 1967). The machine has a single, all-encompassing attribute, its "internal state." The internal state of a finite state machine is described formally as an "equivalance class" of input histories; i.e., a set of input sequences having the common property that for every possible sequence of subsequent inputs, the same sequence of outputs is produced. The internal state of a finite state machine thus uniquely determines its input/output behavior. A key postulate of finite state machine theory is that the number of distinct internal states any given machine can assume is finite.

The possible activities of a finite state machine can be described simply in terms of its input/output properties. Each time an input signal, s, is issued, the machine responds by (1) issuing a corresponding output signal, r; and (2) assuming a new internal state, represented by q.[2] The relationship between an input signal and the corresponding machine response is expressed by defining equations of the following general form:

[2]The symbols used here are independent of those used in defining the parameters in Sections 3, 4, and 5.

Finite State Machine:



Defining Equations:

$$r(t+1) = F\left[s(t), q(t)\right]$$

$$q(t+1) = G\left[s(t), q(t)\right]$$

Defining Tables:

Internal State q(t)

Input Signal s(t)	q_1	q_2	q_3	$\cdots$	q_p
s_1	r_{11}	r_{12}	r_{13}		r_{1p}
s_2	r_{21}	$\cdots$			r_{2p}
s_3					
$\vdots$					
s_m	r_{m1}	$\cdots$			r_{mp}

Output r(t + 1)

Internal State q(t)

Input Signal s(t)	q_1	q_2	q_3	$\cdots$	q_p
s_1	q_{11}	q_{12}	q_{13}		q_{1p}
s_2	q_{21}	$\cdots$			q_{2p}
s_3					
$\vdots$	$\vdots$				
s_m	q_{m1}	$\cdots$			q_{mp}

State Transition

Figure 2.1. Finite state machine fundamentals.

$$r(t+1) = F(s(t), q(t))$$
$$q(t+1) = G(s(t), q(t)).$$

The first equation states that the signal output by the machine at a particular response time, $r(t+1)$, is uniquely determined by a previous input, $s(t)$, and the previous internal state, $q(t)$. The second equation establishes a similar relationship between the internal state assumed at a particular response time and the previous input and internal state. Note that time is treated in a very elementary way in these expressions, as an ordered set of integers which essentially "number" the input and response events in sequence. Time intervals between events are not explicitly considered. The defining equations for a finite state machine can be represented either in tabular form, as shown in Figure 2.1, or (more graphically) in the form of state transition diagrams.

The finite state machine is useful in representing the operation of discrete systems and processes for two reasons. First, the finite state machine is conceptually simple. Its boundaries are clearly defined functionally; its interactions with the environment occur in distinct, discrete events which form a linear sequence in time; and its behavior can be described completely and unambiguously by means of the defining equations. Second, the finite state machine is very broadly applicable. Virtually any physically realizable discrete process can be represented by a single finite state machine from an input/output (black box) point of view; and appropriately defined individual machines can be interconnected in building-block fashion to model the internal behavior of complex systems at whatever level of detail may be required. State diagrams have become the method of choice for defining communication interaction protocols (Bjorner, 1970; Stutzman, 1972; ANSI, 1976; IBM, 1976; CCITT, 1977).

In order to make the finite state machine directly useful as a performance model, we must generalize its description in a number of respects. The following generalizations relate to the definition of machine inputs and outputs:

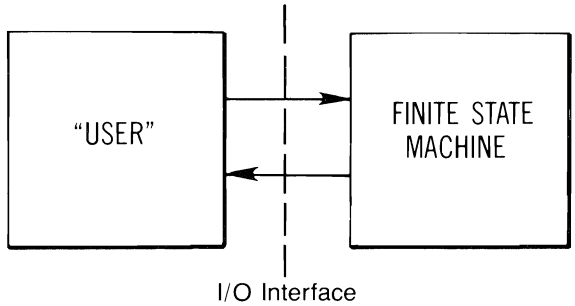
1. Parallel input/output. A set of information units transferred simultaneously over many input or output "channels" can be viewed as a single input or output event.
2. Serial input/output. Any defined series of information units consecutively input to or output from a finite state machine over a single channel can be viewed as a single input or output. Thus, it is possible to view an information block comprised of many serially transmitted bits as a single input to or output from a model entity.
3. Null inputs and outputs. We specifically include null inputs and outputs as valid entries in the state machine defining tables. It is thus possible to define inputs which produce internal state transitions with no corresponding output; and state machine outputs with no corresponding input.
4. Nature of outputs. Finite state machines may produce either static outputs, which are held continuously during a specified state or group of states; or pulsed outputs, which are transferred to the environment at a single, discrete point in model time (IBM, 1976).
5. Control of information input and output. In conventional finite state machine theory, it is implied that the transfer of information into a state machine is controlled by the external environment; and that the transfer of information out of a machine is controlled by the machine. We stipulate that input to or output from a finite state machine can be controlled by either the machine or the environment.

The final generalization we must accept in order to make the finite state machine useful as a performance model is the following. We stipulate that although the intended behavior of the machine is deterministic, as specified by the defining equations, its actual behavior is, at least to some extent, stochastic. Given any initial state q and input signal s, the machine should issue a particular output signal, say r_1. Nevertheless, its actual behavior is such that it may instead issue some other output signal $r \neq r_1$; or it may issue no output signal at all. The outputs of the generalized machine are also variable with respect to time. In a deterministic finite state machine, there is a fixed relationship between the time of occurrence of an input and the time of occurrence of the corresponding output. We stipulate that in the generalized machine, the response to an input can occur at any one of a series of discrete times after occurrence of the corresponding input, up to some maximum waiting time. These modifications make the generalized finite state machine somewhat unpredictable in its behavior, as are all real systems. Our purpose in defining performance parameters is to provide effective means of describing these "real world" departures from ideal behavior in specific, quantitative terms.

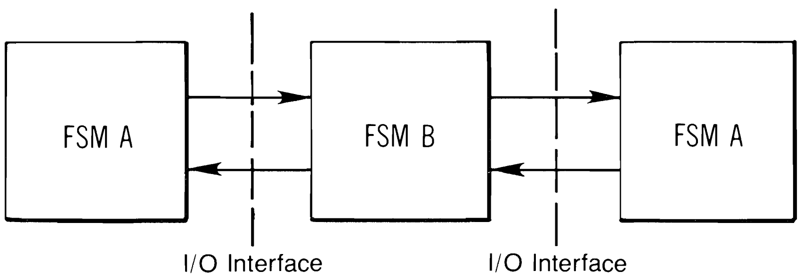
Application of finite state machines in the telecommunication process model will involve the interconnection and joint operation of several machines. Figure 2.2a illustrates the simplest form of interconnection, in which a single finite state machine (FSM) is interconnected with an external environment or "user". The inputs to the finite state machine are (identically) the outputs of the environment; and conversely. If the environment is also viewed as a finite state machine, the two machines comprise a symmetrical pair of performing entities.

Figure 2.2b illustrates the tandem interconnection of three finite state machines. A configuration of this type will be used as basis for the telecommunication process model, with the peripheral machines representing the users and the central machine representing the telecommunication system.

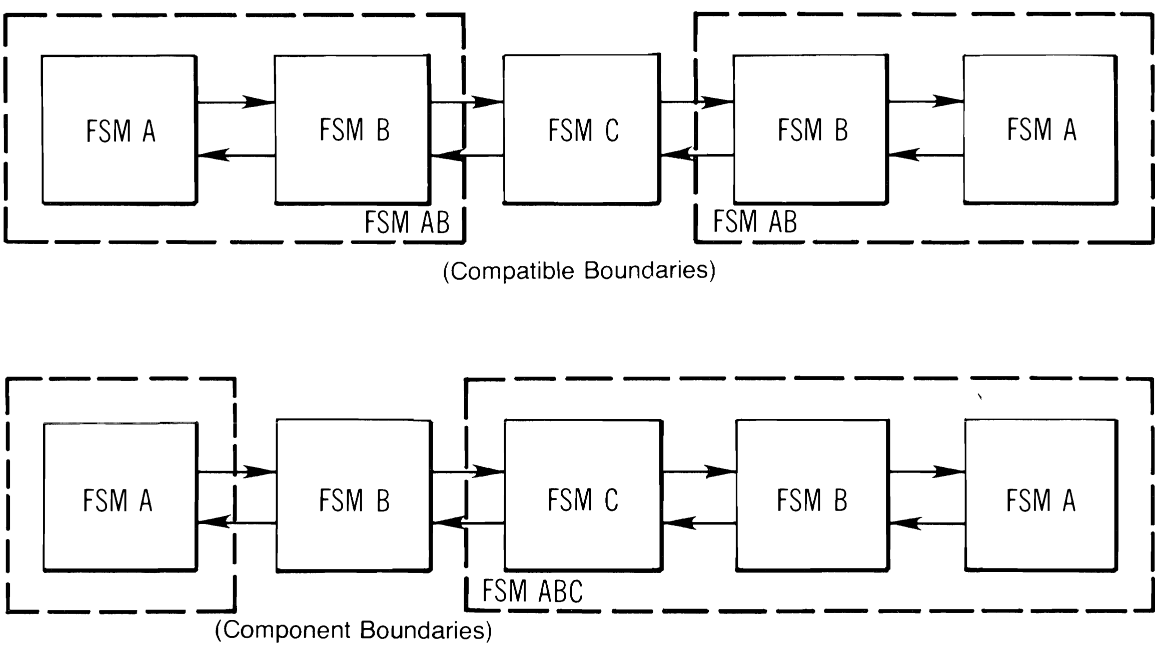
Figure 2.2c illustrates the fact that interconnected finite state machines can be aggregated and treated as a single machine (IBM, 1976). Machines can be aggregated at compatible boundaries, to produce functionally symmetrical performing entities; or at component boundaries, to distinguish a particular physical



a. State Machine with Connected "User"



b. Tandem Interconnection of Three State Machines



c. Aggregation of State Machines

Figure 2.2. Interconnection of finite state machines.

unit or functional entity from its environment. This principle would be used in allocating end-to-end performance requirements to purchasable components or services.

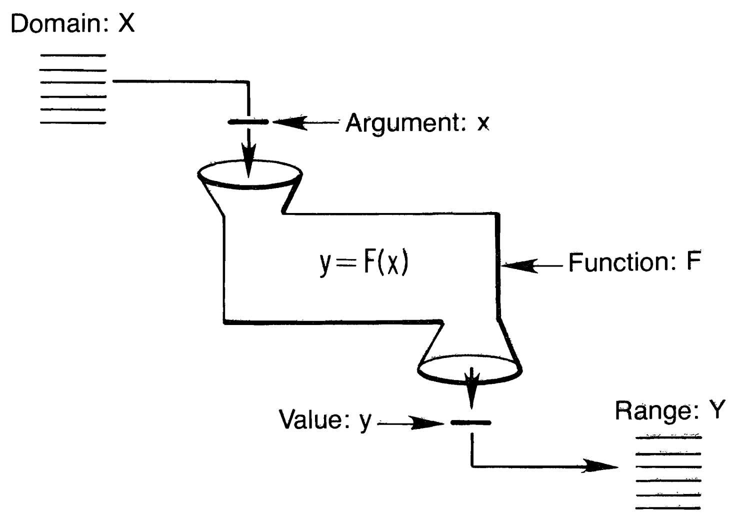
2.2.3. Function Concept

In the following subsection, we propose a general method of distinguishing "the telecommunication system" from "the users" based on differences in the types of functions they perform. Before we can do this, we must first establish a precise meaning for the term "function" as it applies to the performance of a finite state machine. In the mathematical context, a function is typically defined as a set of ordered pairs of elements (x,y) such that one and only one value y corresponds to each possible argument x. The set of possible arguments is termed the domain of the function; the set of possible values is termed the range of function. These terms are illustrated in Figure 2.3a, which portrays the function as a machine which translates arguments into values (Goldberg, 1960).

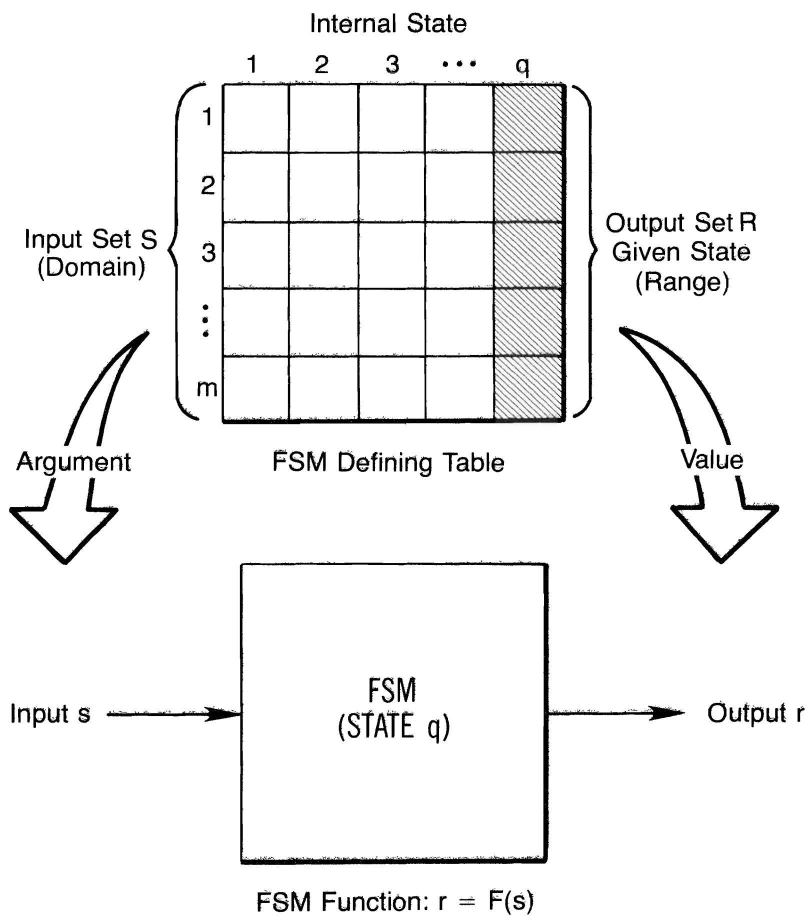
The concept of a function as a machine leads naturally to the view that the finite state machine can be used to represent a function; and in the context of operation of a state model, to perform a function. In the following, we define a function of an individual finite state machine in the mathematical sense, as a set of input/output pairs (s,r) such that one and only one output corresponds to each input (Fig. 2.3b). In general, then, each column of an FSM input/output defining table specifies a function the machine can perform.

2.3. User/Telecommunication System Functional Interface

In discussing data communications, we refer frequently to the terms "user," "telecommunication system," "functional interface," "user information," "overhead information," "telecommunication service," and "information transfer transaction." The purpose of this subsection is to define these terms as precisely as possible, as they are used in connection with the proposed Federal Standard; and to provide a set of ground rules for identifying appropriate



a. Mathematical Definition of a Function



b. FSM Performance Adaptation

Figure 2.3. Definition of FSM functions.

end-to-end performance assessment interfaces in any particular system.

The performance assessment concepts introduced above are readily defined in the case of most traditional data communication services. The user is either a human terminal operator or a non-electromagnetic medium supporting his information storage needs (e.g., punched cards or paper tape). The telecommunication system consists of the data terminals, modems, and analog transmission facilities which interconnect user entities. The user/ telecommunication system functional interface is at the physical interface between the data terminal and the operator or medium on its drop side. The data terminal is thus regarded as a part of the telecommunication system from the standpoint of user-oriented performance assessment. User information consists of all digital information which crosses both user/system interfaces. All other transferred information (i.e., USASCII ENQ, ACK, and SYN characters) is defined as overhead information. The system provides a telecommunication service to the users by transferring digital user information between them; the sequence of user/system interactions by which this service is provided in a particular instance is called an information transfer transaction.

The difficulty in defining the above concepts (and the primary need for the telecommunication process model) arises when we consider the case of computer communication and teleprocessing networks. There may be no distinct physical unit which corresponds to the data terminal in such networks; instead, the data terminal functions of data segmentation, error control, etc. are performed by software programs within each host computer. Complete information transfer transactions can take place in such networks with no human operator or non-electromagnetic medium involved. Typically, "the user" who initiates such transactions is a computer applications program; and the transaction is controlled by "system calls" or "macro instructions" exchanged between that program and a telecommunication access program ("access method") within the same host. The "access method" thus represents the termination of the end-to-end telecommunication system in a typical "layered" teleprocessing network (e.g., see McFadyen, 1976).

The terms "applications program" and "telecommunications access method" are widespread enough that they could be used as the basis of defining user/system interfaces in most modern computer/communication and teleprocessing networks (e.g., see IBM, 1976; Computer News, 1977; Schwager, 1978). Nevertheless, there are situations in which these terms alone do not provide an adequate interface definition. The terms "application program" and "access method" are not even used in the literature describing some computer communication networks; and in many older "non-layered" teleprocessing systems, clearly communication-related functions such as polling and switching are performed by what are called "applications programs."

These considerations suggest the need for a general method of establishing end user/telecommunication system interfaces, based on differences in the functions performed by the various participating entities (e.g., computer programs). The following paragraphs describe such a general method. The method is intended for use in establishing Federal Standard 1033 performance assessment interfaces wherever the above more specific criteria are inapplicable. Although the general method is needed primarily in "exceptional" cases, as described above, it can be used to establish appropriate end-to-end interfaces in any digital communication configuration. It is emphasized that the user/system interfaces established by this method will not always coincide with the physical common carrier/customer interfaces. Further, the method makes no assumptions about what elements of an end-to-end telecommunication service may be subject to regulation; this issue, currently being debated under the FCC's second Computer Inquiry (FCC, 1976) is outside the scope of both this report and the proposed Federal Standard.

In developing the general method, we make use of the finite state machine as a descriptive tool for representing any functional element (entity) involved in an information transfer transaction; and define the term "function" in the mathematical sense, as described in Section 2.2. We begin by considering the various types of interactions that can take place between an individual finite state machine and its external environment, here regarded as a "user" of the machine's "service" (Fig. 2.2a). From the point of view of the environment, there are two general cases of interest. The first is the case in which the internal state of the FSM is known to the environment, and is appropriate to its "intended use" of the machine. Whenever these conditions hold, a unique machine output is associated with each input, and the machine can perform a useful function for the environment. Two general classes of functions can be distinguished:

1. Functions in which a single, unique input is associated with each output. For all input/output pairs in this category, the input is uniquely derivable from the output; the purpose of such functions must therefore be either to transfer the input information to the output, or to change its form of representation. We define such functions as data communication functions[3].
2. Functions in which more than one input is associated with each output (note that there is still one unique output associated with each input). For all input/output pairs in this category, the input is not uniquely derivable from the output; instead, the input data has been "processed" to produce some new information. Since the purpose of such functions must be to process the input data in a useful way, we define this class of functions as data processing functions.

For both of these classes, the purpose of input to the FSM is to obtain the machine's service in performing the specified function.

---

[3]Such functions are described as "one-to-one onto mappings" in the formal literature of discrete mathematics (e.g., see Stone, 1973).

The distinction between data communication and data processing functions can be stated somewhat more precisely in terms of the entropy concept from information theory (Shannon, 1948). The data communication functions defined above are functions whose intent is to preserve the entropy of the source (i.e., transfer information from source to destination without introducing uncertainty or "equivocation"). The data processing functions defined above are functions whose intent is to reduce the entropy of the source in a defined way, making the input information more structured, more predictable, more ordered, and less random. The concept of "information content" or "meaning" is thus associated with source entropy, which is quite precisely defined in information theory. We note that a data communication function can be "nested" within a data processing function, since the data communication functions are entropy-preserving; but the converse is not true.

The second case of interest is the case in which the FSM state either is not known to the environment or is not appropriate to its intended use of the machine. In this case, the machine cannot perform a useful function for the environment. The purpose of interaction between the two entities under these conditions must be to place the machine in the appropriate (intended) state, and to confirm the existence of this state to the environment. Our interest in interactions of this type is restricted to interactions which change the machine's ability to perform the function of data communication, as defined above. We define such interactions as communication processing.

Strictly speaking, the communication processing interactions are not functions, since (in general) there is no unique machine output associated with each user input. We will nevertheless refer to these interactions as functions, to avoid introducing a new term. (Mathematicians do sometimes describe input/output sets of this type as "multi-valued" functions). Examples of communication processing functions performed in digital communication systems are physical and virtual circuit establishment, error control, and polling.

In order to make the general method of establishing interfaces applicable in all cases, we must extend the above definitions in one additional way. We define two subclasses of data communication functions: telecommunication functions and non-electromagnetic communication (NEMC) functions. Telecommunication functions are data communication functions in which the digital information is stored and transported by electromagnetic means. NEMC functions are data communication functions in which the digital information is stored or transported by non-electromagnetic means; e.g., physical transport of "hard copy". This distinction is consistent with the CCITT definition for telecommunication (CCITT, 1973):

> Telecommunication. Any process that enables a correspondent to pass to one or more given correspondents (telegraphy or telephony), or possible correspondents (broadcasting), information of any nature delivered in any usable form (written or printed matter, fixed or moving pictures, words, music, visible or audible signals, signals controlling the functioning of mechanisms, etc.) by means of any electromagnetic system (electrical transmission by wire, radio transmission, optical transmission, etc., or a combination of such systems).

A third type of data communication function that can be defined is the function of media conversion. In the following, we restrict this concept to the conversion of information between electromagnetic and non-electromagnetic form; and view media conversion as a special type of telecommunication function. This is also consistent with the CCITT definition just cited. We thus have four distinct classes of functions that can be performed by an individual finite state machine for its environment (Fig. 2.4a):

1. Telecommunication (TC) functions
2. Non-Electromagnetic communication (NEMC) functions
3. Data processing (DP) functions
4. Communication processing (CP) functions.

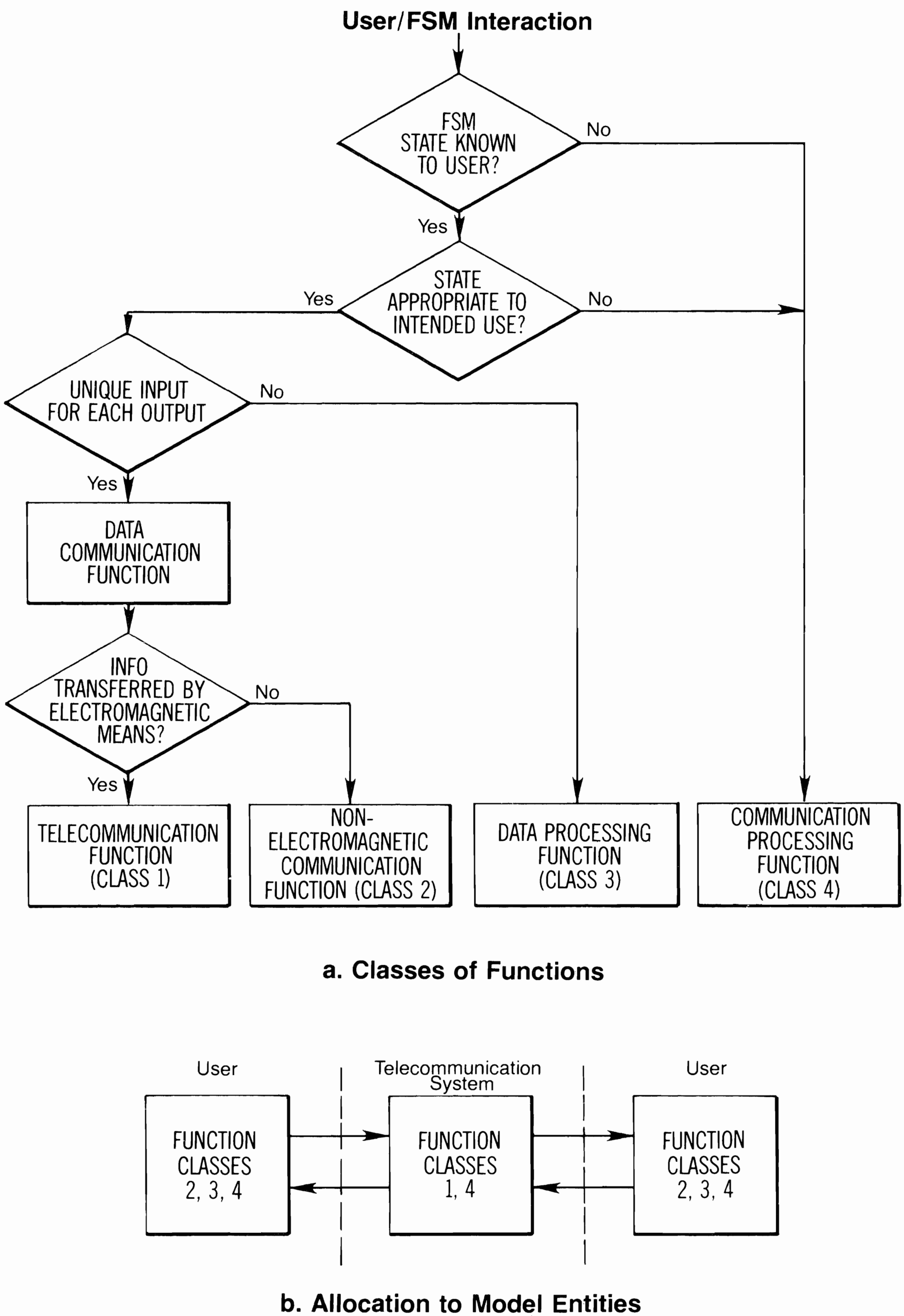
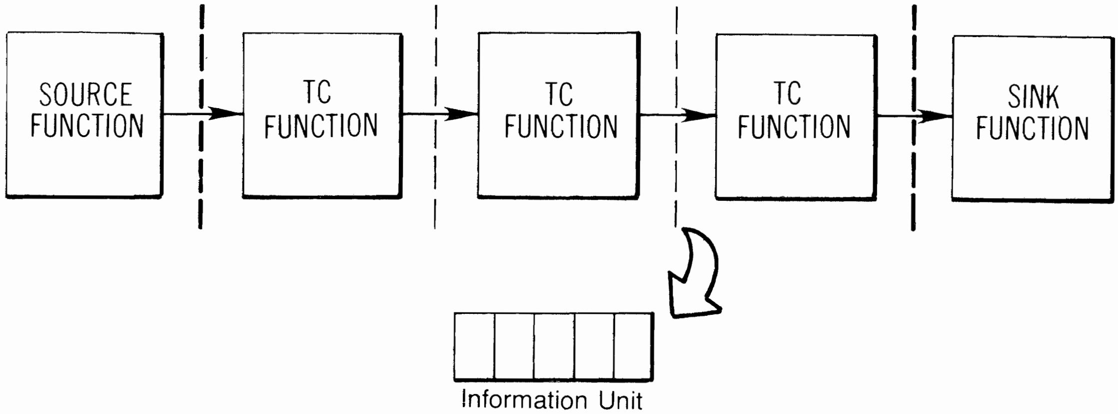


Figure 2.4. Functional definition of model entities.

The scope of the telecommunication process model can now be described in terms of these four classes of functions. The telecommunication process model is intended to represent interactions between user and system entities represented by finite state machines which are interconnected and functionally constrained as shown in Figure 2.4b. Specifically, the two peripheral entities ("users") are constrained to perform functions of types 2, 3, and 4; they perform no telecommunication functions (except for "nested" functions of the type described above). The central entity ("the telecommunication system") is constrained to perform functions of types 1 and 4; by definition, it performs no data processing or NEMC functions. We state that the system provides a <u>telecommunication service</u> to the user entities by transferring information between them; and define the process by which this service is provided in a particular instance as an <u>information transfer transaction</u>.

It is emphasized that the constraints defined above place no restrictions on the functions of physical facilities; they simply specify the kinds of functions that are relevant to the telecommunication process model. For example, a host computer in a teleprocessing network typically performs data processing, telecommunication, and communication processing functions for the local terminal operators. If such a computer were included within the system portion of the telecommunication process model, only the latter functions would be considered relevant.

We now propose a method of using the function categories described above to locate the user/system interfaces in any particular communication configuration. Imagine a chain of functions, performed in one or more physical units, as illustrated in Figure 2.5a. Consider a unit of information at any point in the chain, at the output of one function and the input of the next. If the function outputing the information is a telecommunication function, we can (conceptually) trace the information backward to the function input, where it will continue to exist; we can continue this process, tracing the infor-

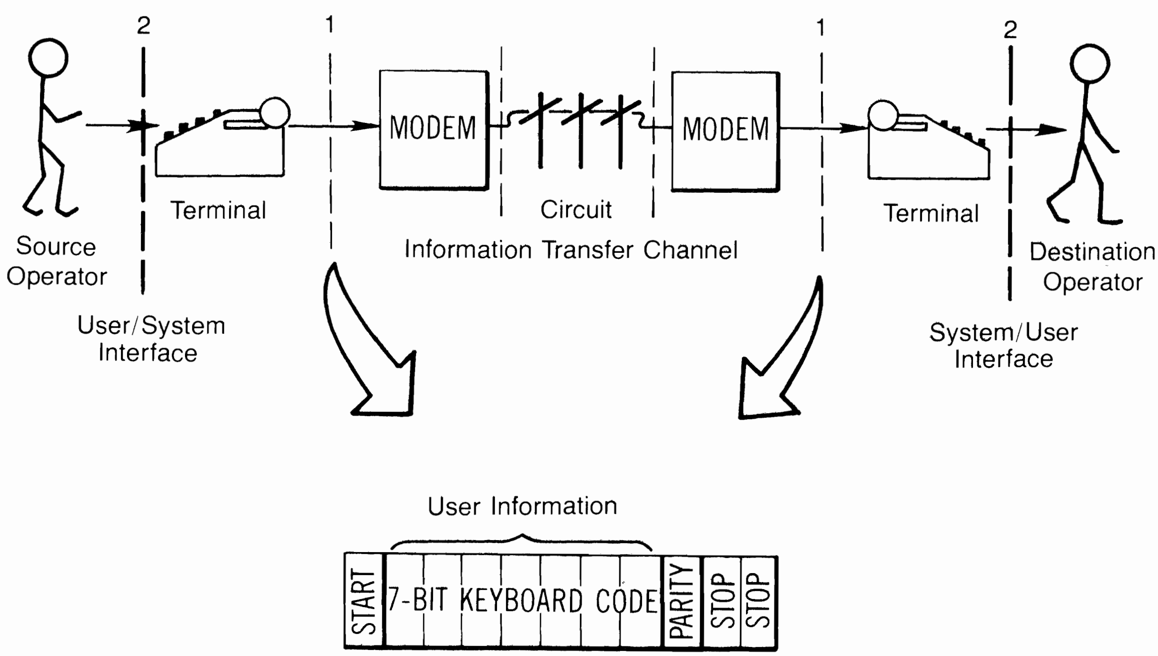


<u>User Information</u>:
Source <u>and</u> Sink Functions = DP or NEMC

<u>Overhead Information</u>:
Source <u>or</u> Sink Function ≠ DP or NEMC

<u>User/System Functional Interfaces</u>:
- Output of Source Function that Produces User Information
- Input to Sink Function that "Consumes" User Information

a. General Definitions



b. TTY Example

Figure 2.5. User/system interface definitions.

mation "upstream", until we reach a point where the information arises in electromagnetic form, in one of three ways:

1. As output from an NEMC function (i.e., input to a telecommunication function of media conversion);
2. As output from a data processing function;
3. As output from a communication processing function.

In a similar way we can (conceptually) trace any information unit "downstream" to the point where the information ceases to exist in electromagnetic form. Here again, the information will "disappear" in one of three ways:

1. As input to a data processing function;
2. As input to an NEMC function (i.e., output from a media conversion function);
3. As input to a communication processing function.

We defined user information and overhead information in terms of source function and sink function as follows:

Source Function	Sink Function	Information Type
DP	DP	User
DP	NEMC	User
DP	CP	Overhead
NEMC	DP	User
NEMC	NEMC	User
NEMC	CP	Overhead
CP	DP	Overhead
CP	NEMC	Overhead
CP	CP	Overhead

User information is defined as information for which both the source function and the sink function are either data processing or non-electromagnetic communication. Overhead information is defined as information for which either the source function or the sink function is communication processing. A user/telecommunication system functional interface exists at the output of any source function that produces user information; and at the input to any sink function that "consumes" user information.

These general guidelines are consistent with the specific interface definition criteria defined earlier for traditional data communication services, as Figure 2.5b illustrates. The functional chain in this example is comprised of five primary entities: the two terminal operators; two data terminals (e.g., ASCII teletypewriters); and an information transfer channel, comprising two modems (DCE's) and a communication circuit. Each character that crosses the DTE/DCE interface consists of a start bit, seven bits which represent the keyboard character entered by the transmit terminal operator, a parity bit, and two stop bits. Two possible approaches could be used to define the telecommunication system boundaries in this configuration:

1. Define the boundaries at the traditional DTE/DCE interchange points (points 1 in Fig. 2.5b); or
2. Treat the terminals as part of the telecommunication system; and define the telecommunication system boundaries on the drop side of the data terminals (points 2 in Fig. 2.5b).

Only the latter approach is consistent with the definitions presented above, since only the seven-bit keyboard code qualifies as user information; and this information has its source and sink in the terminal operators. The terminals clearly perform the function of telecommunication on the keyboard bits.

Although the above guidelines are useful in establishing end-to-end performance assessment interfaces, their application requires some judgment: they do not address the level of detail to which tandem functions should be resolved. As an example, common carrier technical references commonly use the term "station" to denote either the operator or data terminal, implying that the operator and terminal are a single functional entity which "uses" the information transfer channel. This coarse resolution of functional entities would be appropriate (e.g.) for allocating a portion of an end-to-end delay or error "budget" to the information transfer channel; but it would not be appropriate for establishing the basic end user interfaces.

In the final analysis, the level of detail to which tandem functions must be resolved to establish appropriate user/system interfaces can only be addressed by example. Nevertheless, the following principles will be helpful:

1. Any physical interface should be regarded as a functional boundary. Any computer program interface within a teleprocessing system should also be regarded as a functional boundary.
2. Error control, flow control, and polling should be distinguished as separate functions irrespective of physical equipment boundaries.
3. Any distinct protocol boundary (e.g., link control, virtual circuit, path control, etc.) should be viewed as a functional boundary, whether it occurs between physical units or within a single physical unit.
4. Communication processing functions associated with the user should be limited to (a) requesting an information transfer service; (b) providing overhead information needed to identify the intended source and destination, to identify and delimit the user information, and to report user status; and (c) requesting termination of an established transaction.

The guidelines presented above are basically compatible with the way both AT&T and IBM define user/telecommunication system interfaces. Representatives of the Bell System have described data communications as being comprised of three primary elements: transmission, communication processing, and media conversion (Datamation, 1976). IBM has defined the "end user" of its Systems Network Architecture (SNA) network as being one of three entities: a human terminal operator, an applications program, or an unattended device medium such as punched cards (IBM, 1976). In both cases, "the telecommunication system" extends to the drop side of the data terminal or equivalent function.

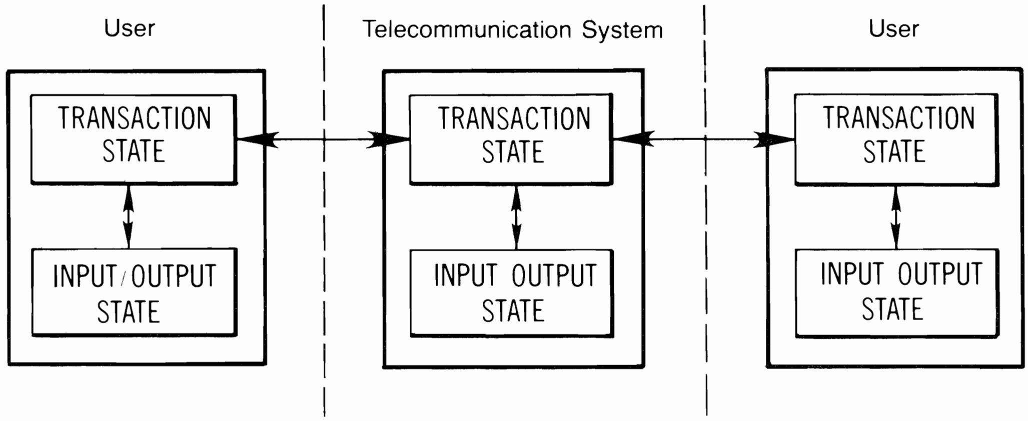
2.4. The Model

The preceding subsection defined three model entities, each represented by a finite state machine having specific functional capabilities. This subsection completes the development of the telecommunication process model by defining the specific attributes and activities of each entity. The purpose of the model is to provide a uniform basis for defining the functions, outcomes, and parameters specified in the proposed Federal Standard; and to provide a uniform data recording format for use in performance measurement.

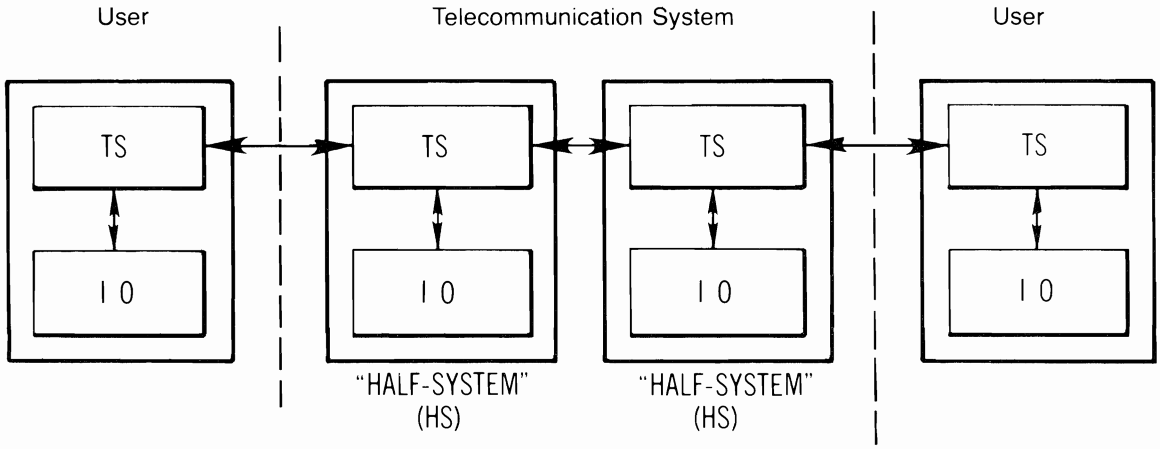
Figures 2.6, 2.7, and 2.8 present the essential details of the telecommunication process model in increasing detail. Figure 2.6a distinguishes the users and the telecommunication system as separate entities, each having as attributes two types of state variables: Transaction state and Input/Output state. The Transaction state variables represent the state of each entity relative to the communication processing functions. The Input/Output state variables represent the state of each entity relative to the system function of telecommunication. The functions of data processing and non-electromagnetic communication exist in the users, but are not included in the model since measurement of these functions is outside the scope of the standard. From the point of view of the model, these functions are simply mechanisms by which user information is produced and "consumed."

The arrows between entities in Figure 2.6a represent discrete transfers of user and overhead information. These transfers ("interface events") result from the telecommunication and communication processing activities; and are recorded by appropriate Transaction and Input/Output state changes. At this level, the model is very similar to that proposed by Gray (1972).

Figure 2.6b shows the same general model with the system divided into two "half-system" entities, each associated with one of the two user interfaces. This division reflects the fact that the system activities underway at one user interface



a. Three-Entity Model

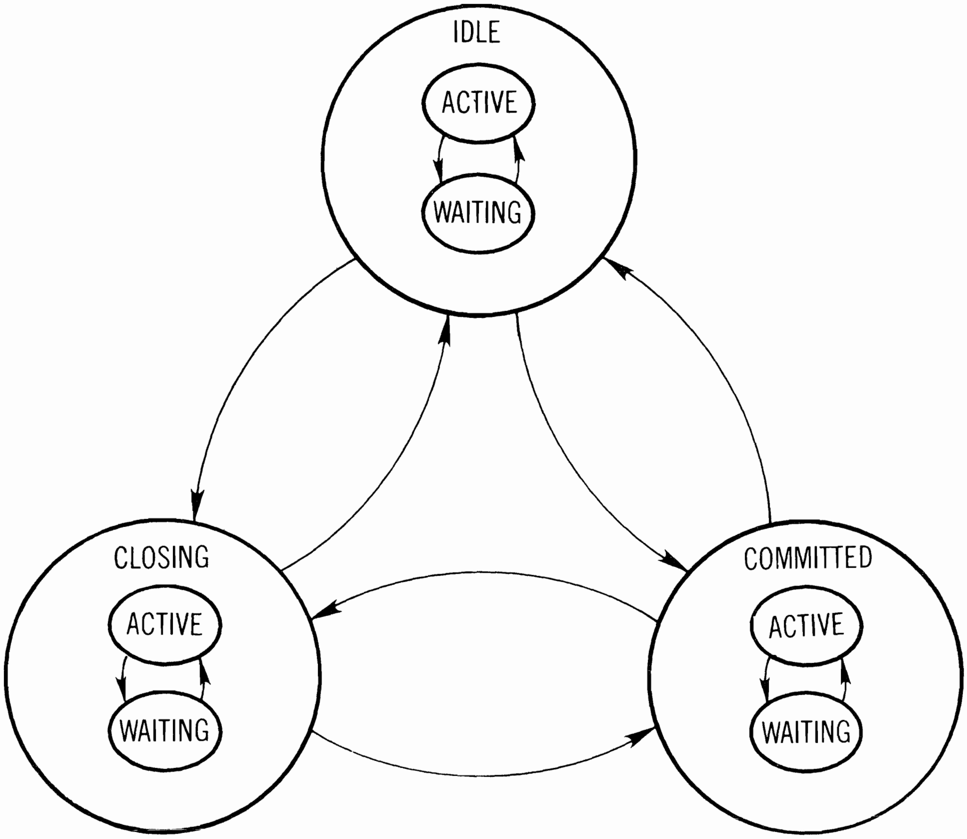


b. Four-Entity Model

EVENT NUMBER	EVENT TIME	EVENT SENSE	ENTITY STATE TRANSITIONS			
			USER	HS	HS	USER
1	—	—	—	—	—	—
2	—	—	—	—	—	—
3	—	—	—	—	—	—
⋮						⋮
n	—	—	—	—	—	—

c. State History Format

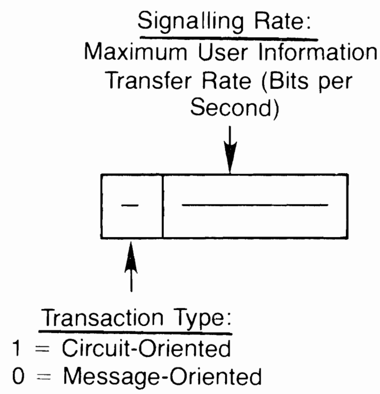
Figure 2.6. Model overview.



a. Entity Transaction State Diagram

COMPOSITE STATE	CLOSING STATE	COMMITTED STATE	RESPONSIBILITY STATE
IDLE/ACTIVE	0	0	1
IDLE/WAITING	0	0	0
COMMITTED/ACTIVE	0	1	1
COMMITTED/WAITING	0	1	0
CLOSING/ACTIVE	1	0	1
CLOSING/WAITING	1	0	0

b. Binary Representation of Transaction States



c. System Property Vector

Figure 2.7. Transaction state and system property variables.

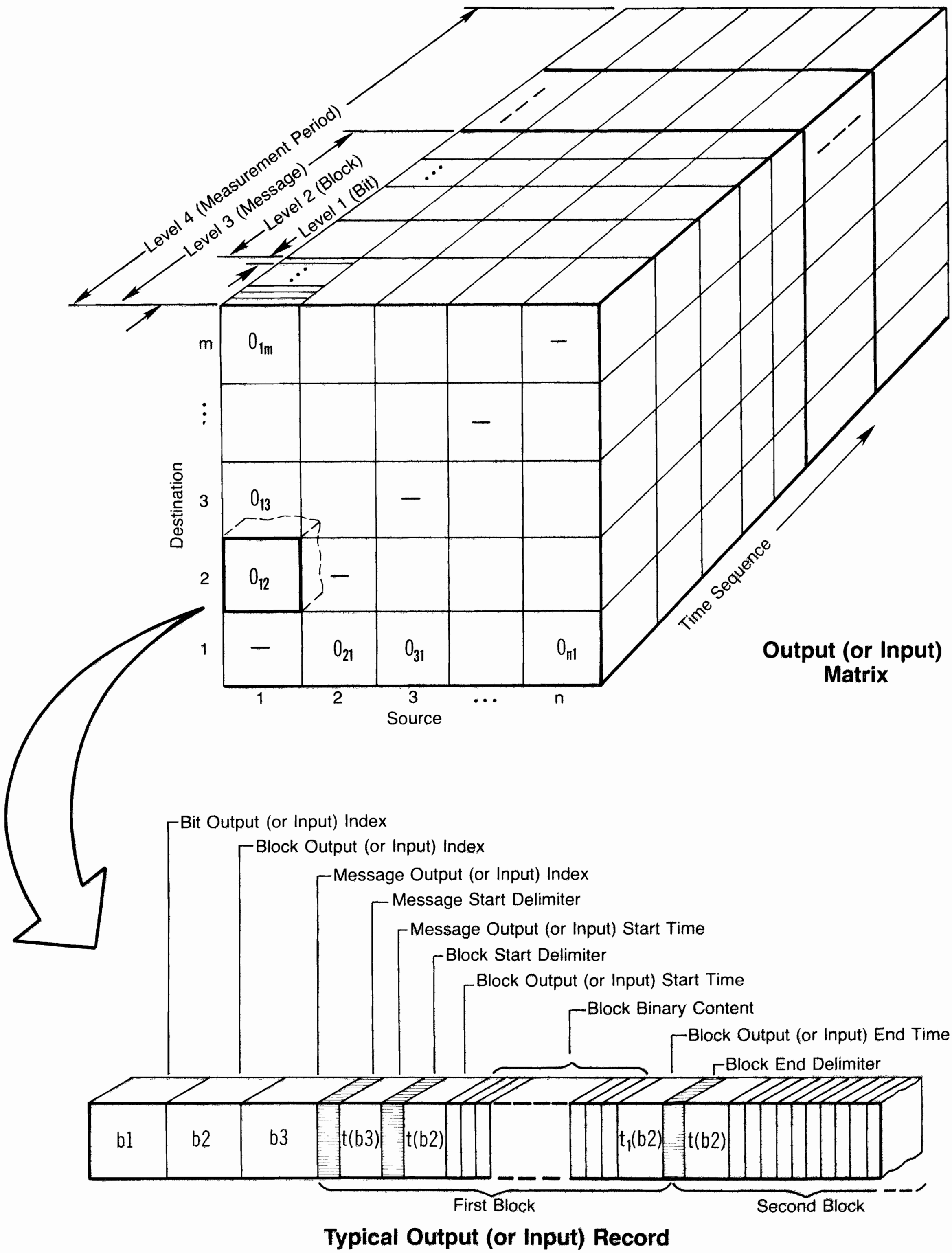


Figure 2.8. Format of output (or input) record.

may be completely uncorrelated with those underway at the other interface during a portion of an information transfer transaction; different system Transaction states may thus apply to each interface. Note that the total state of the telecommunication system can still be represented, as a concatenation of the two half-system states. We still define only two interfaces in the model, one associated with each user.

Figure 2.6c shows the general form of a model state history. The state history comprises a linear sequence of interface events and associated entity state transitions. Each interface event consists of the transfer of a defined unit of overhead or user information from the physical possession and control of one functional entity (user or half-system) to that of the "opposite" entity (half-system or user). Each is characterized by two general properties: a discrete event time, which indicates when the associated information transfer took place; and an event "sense," which indicates the direction of the associated transfer (i.e., user-to-system or system-to-user). "Sense" information is actually needed only in the case of overhead information transfers.

The Transaction state and Input/Output state variables record the nature of each interface event from the standpoint of performance assessment. Overhead information transfers produce changes in the Transaction state variables; user information transfers produce changes in (additions to) the Input/Output state variables. All user information output by an entity during a performance measurement period is recorded in its Output state variable; all user information input to an entity during a performance measurement period is recorded in its Input state variable. Telecommunication performance is measured by analyzing such a state history, over a suitable performance measurement period, to define telecommunication functions, performance outcomes, and performance parameters.

Since each user/system interaction is recorded by an associated model state transition, it is possible to define communication <u>functions</u> directly in terms of model state transitions rather than in conventional input/output terms. In general, any pair of model state transitions which necessarily occur in a specified order can be used to define a communication function. There is no requirement that the state transitions must correspond to inputs and outputs of the same model entity; or that they must be immediately adjacent in the model state history. Transitions which necessarily occur between the starting and ending events defining a function are termed "intermediate events."

Figure 2.7a is a state diagram which represents the possible Transaction states of each of the four model entities (the two users and the two half-systems). Although the Transaction state variables of the four entities are closely linked, a separate variable applies to each entity, and no one variable completely defines another.

Relative to any given transaction, each entity is in one of three primary Transaction states at any time. These three states are defined as follows.

1. Idle state. The entity is not involved in the subject transaction. The entity can be involved in another transaction, or can be uninvolved in any transaction.
2. Committed state. The entity is involved in the subject transaction, with intent to transfer further user information.
3. Closing state. The entity is involved in the subject transaction, with intent to terminate involvement without transferring further user information.

All transitions between primary Transaction states occur as a result of overhead information transfers. Overhead transfers normally cause identical primary Transaction state transitions in both communicating entities. As an example, transfer of an access request (e.g., off-hook signal) between two entities in the Idle state normally causes both entities to enter the Committed state; transfer of a disengagement request (e.g., Close Request message) between two entities in the Committed state normally causes both entities to enter the Closing state.

The users are described as originating or non-originating users based on the order of commitment at the two user interfaces. The user that first enters the Committed state during a transaction is defined as an originating user for that transaction; and conversely.

Within each primary state are two "ancillary states": the Active state and the Waiting state. In general, an entity can enter either ancillary state on entry to the corresponding primary state; and can change ancillary states any number of times within a given primary state. The ancillary substates have slightly different meanings depending on the associated primary state. Within the Committed and Closing states, the terms active and waiting describe an entity's "responsibility" relative to the ongoing sequence of interactions at the local user interface. When an entity is responsible for producing the next interface event at the local interface, the entity is defined to be in the Active state; otherwise, the entity is defined to be in the Waiting state.

All substate transitions within the Committed and Closing states occur as a result of overhead information transfers; but not all such transfers produce responsibility state changes. The direction of responsibility transfer between communicating entities is not necessarily the same as the direction of the associated overhead information transfer. The two directions differ, for example, in the case where a data terminal (half-system) actively reading a punched tape of user information encounters a "stop code." This event transfers responsibility for continuing user information transfer from the system to the user; i.e., the user must load a new tape or terminate the transaction.

Many telecommunication services are available to the user only during certain designated time periods, termed "service time intervals." Users may restrict their participation in telecommunication activities in a similar way. The Active and Waiting substates within the Idle state represent the status of an entity relative to such service restrictions. When an entity is within a designated service time interval, but is not involved in an information transfer transaction, the entity is defined to be in the Idle/Active state. When an entity is not within an established service time interval, the entity is defined to be in the Idle/Waiting state. Transitions involving these states indicate the beginning and end of planned service time intervals. Note that it is possible for two half-systems within the same telecommunication system to be in different Idle substates; an example is a world-wide message switching system that provides service to subscribers only during local business hours.

Figure 2.7b represents the six Transaction states defined above in three-bit binary form. Various constraints on the joint holding of ancillary states by the four model entities are described in Section 5.

Figure 2.7c defines a System Property vector which is an accessory to the Transaction state variables of the two half-systems. The same System Property vector applies to each half-system. The vector is fixed for any given telecommunication service; it specifies those system characteristics which influence end user assessment of performance.

The first element of the System Property vector, Transaction Type, distinguishes two fundamental types of end-to-end communication services:

<u>Circuit-Oriented</u> - both users must be in the Committed state before the transfer of user information can begin.

<u>Message-Oriented</u> - user information transfer can begin with only one of the two user entities (the source of the user information) in the Committed state.

This system property is used in determining access outcomes.

The second element of the System Property vector is Signalling Rate (R_{max}). Signalling Rate is defined as the maximum rate, in bits per second, at which binary information could be transferred (in a given direction) between users over the telecommunication system facilities dedicated to a particular information transfer transaction, under conditions of (1) continuous transmission, and (2) no overhead information. For a single channel, Signalling Rate is expressed as

$$R_{max} = \frac{1}{\tau} \log_2 \eta$$

where η is the number of significant conditions of modulation of the channel and τ is the minimum time interval (in seconds) for which each level must be maintained. In the case where an individual end-to-end telecommunication service is provided by parallel channels, Signalling Rate is expressed as

$$R_{max} = \sum_{i=1}^{w} \frac{1}{\tau_i} \log_2 \eta_i$$

where w is the number of parallel channels, τ_i is the minimum interval for the ith channel, and η_i is the number of significant conditions of modulation for the ith channel. In the case where an end-to-end telecommunication service is provided by tandem channels, the end-to-end Signalling Rate is defined as the lowest signalling rate among the component channels[4]. Use of Signalling Rate as a normalization constant permits assessment of system transmission resource utilization.

Values for the System Property variables do of course vary from one telecommunication system to another. Nevertheless, their presence in the model does not restrict its application, or make the performance parameters defined on it system dependent. All systems can be described by some (specific) System Property vector; and the same performance parameters apply to all systems, irrespective of System Property values.

The Input/Output state variables constitute the second major component of the state model of each entity. These variables record the occurrence of interface events in the same general way as the Transaction variables, except that the Input/Output variables can assume a much larger number of states. These variables are nevertheless finite, since each measurement

[4]Definition adapted from the CCITT Green Book (CCITT, 1973).

period comprises a fixed maximum number of user information bit transfers. Note also that the presence of Input and Output variables in the user entities does not imply the need for permanent user information storage in these entities as a condition of operation; such storage is needed only to support performance assessment.

The lower portion of Figure 2.8 shows the form of a typical Output state variable. The variable comprises a delimited, time-tagged record of all user information output by a particular source user for delivery to a particular destination. The information is represented as a succession of user-delimited blocks of bits, each associated with a particular multi-block "message" of fixed length. (The latter information unit is basically a sample size, defined only for performance assessment purposes. The rationale for its selection is presented in Section 3.2.1.) Each block and message in the Output record has an associated start and end time; these refer to the start and end of output from the source. Each Output record also includes three output indices, b1, b2, and b3. These indicate the total numbers of bits, blocks, and messages currently represented in the Output record. They are incremented at the start of output of each successive information unit.

The Output variable just described represents only the outputs associated with a particular source/destination pair. The upper part of Figure 2.8 shows how a group of such records would be combined to produce an aggregate Output record for a population of users. Each pair of users is represented by a particular row/column intersection in the matrix. Each column contains all information output by a particular source user, irrespective of destination; each row contains all output information intended for delivery to a particular destination, irrespective of source. A sequence of output data can be examined at various levels of detail, as illustrated in the upper left corner of the figure.

There is an obvious correspondence between the Output state variable of a source user and the Input state variable of the adjacent half-system. To simplify the model, we define these variables as identical, and normally refer only to the variable associated with the user entity.

The Input state variable of a destination user and the Output state variable of the local half-system are also defined as identical variables, represented by the former. The structure of this variable is identical to that of the source user Output variable, except that the time tags and indices refer to destination user input, rather than source user output. The bit, block, and message indices are incremented on completion of input of each successive information unit. It is not assumed that properly associated source user output and destination user input records will be identical; either the content of the destination user input data or the identity of the destination user receiving a transmitted block may differ from that intended by the source.

The phrase "properly associated source user Output and destination user Input records" requires some explanation. In order to use the model in defining telecommunication performance parameters, we must assume the feasibility of a "standard data correlator" capable of matching transmitted and received bits and identifying "undelivered" and "extra" bits. No algorithm has been selected to perform this matching function as yet. Two general approaches to the development of such an algorithm can be considered:

1. Where the time delay between output of an information unit at the source and input of the corresponding unit at the destination is predictable, time tags can be used to associate the two records.
2. Matching can be accomplished by analysis of the data itself.

The first approach is obviously limited to predictable delay systems. The second approach can always be used, in principle; however, it may place constraints on the content of the transferred data. It is an objective of NTIA's Data Communications program to develop a standard data correlator based on the second approach, to be used in conjunction with the measurement standard. This is clearly an important goal if the standard parameters are to be used in comparing the performance of different systems: performance parameter values calculated on the basis of different correlation algorithms would not be directly comparable.

All user information in the Input and Output records can be associated with a particular information transfer transaction by comparison of event times. The model does not require that the flow of user information be unidirectional during an individual transaction; communication can also be half or full duplex. In the latter case, both directions of transfer would be considered in specifying and measuring performance.

2.5. General Application Illustrations

The purpose of this subsection is to illustrate the general operation of the telecommunication process model in representing user/system interactions. Two specific transaction sequences are described: a voice telephone call between two subscribers of the public switched network; and the sending of a telegram between the same two subscribers via public message telegram service. Although these are predominately analog rather than data communication services, they have the merit of familiarity; and this serves to focus attention on operation of the model. The two sequences exemplify the two classical approaches to the provision of switched telecommunication service: circuit switching and message switching. The differences in user/system interactions between these approaches are very significant from the standpoint of standard performance description.

2.5.1. Voice Telephone Transaction

The following steps describe the user/system interactions which occur during a normal voice telephone call in terms of the telecommunication process model of Section 2.4. The overall transaction is illustrated step by step in Figure 2.9.

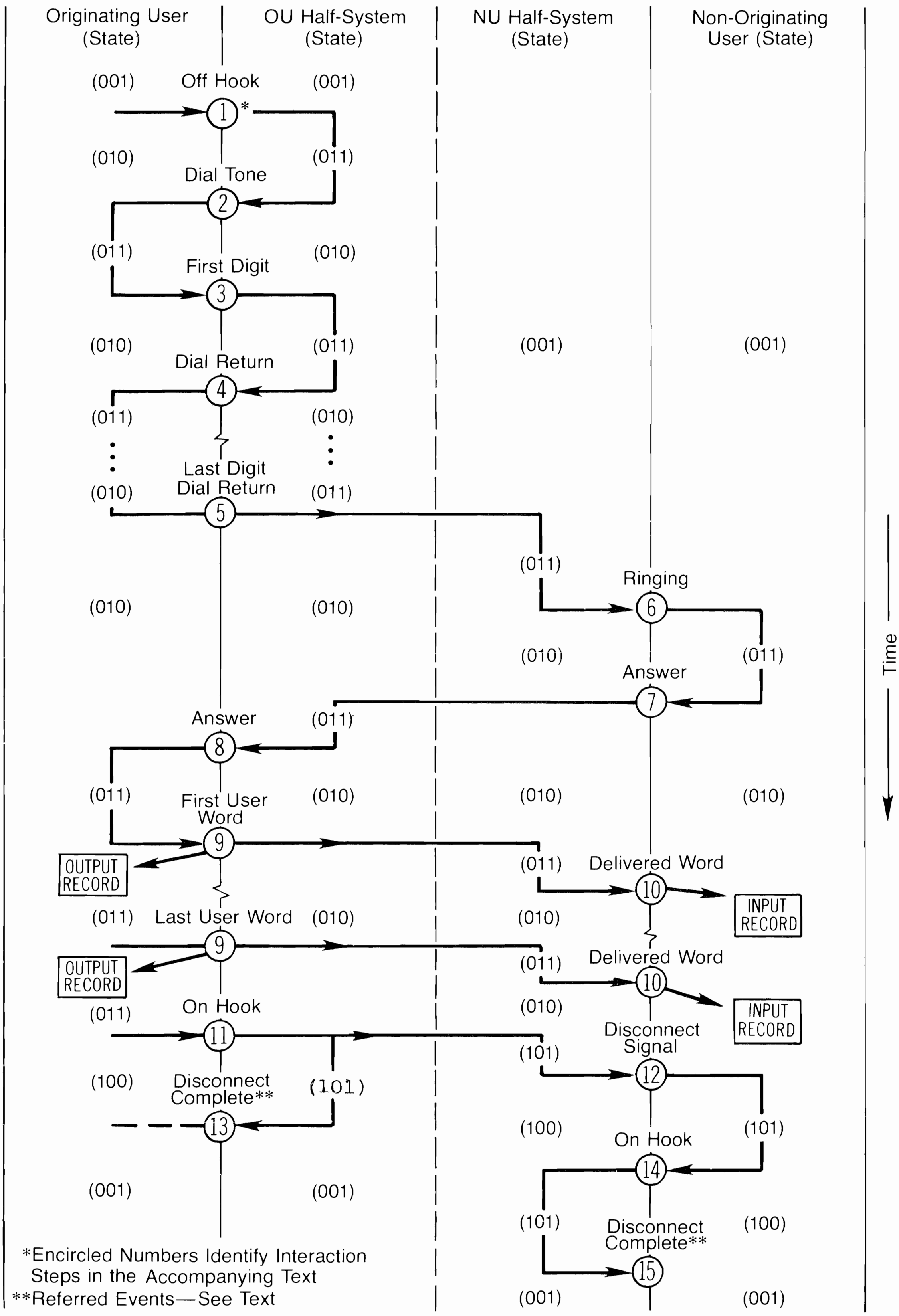


Figure 2.9. State model representation of a voice telephone call.

1. The originating user (subscriber initiating the call) begins the transaction by going off-hook. In terms of the model (Fig. 2.7), this interface event causes the Transaction state of the originating user (OU) to change from Idle/Active (001) to Committed/Waiting (010); and causes the Transaction state of the OU half-system to change from Idle/Active to Committed/Active (011). This responsibility relationship (user waiting, system active) reflects the fact that the system is responsible for producing the "next event" in the transaction sequence.
2. The telecommunication system (specifically, the line scanner at the local central office of the originating user) detects the off-hook condition, commits a dial receiver to the calling line, and issues a dial tone signal to notify the OU that he can proceed to dial the desired number. Issuance of dial tone causes the Transaction state of the OU half-system to change from Committed/Active to Committed/Waiting; and causes the Transaction state of the OU to change from Committed/Waiting to Committed/Active. These state transitions reflect the fact that the OU is now responsible for producing the next interface event.
3. The originating user rotates the telephone dial to a position corresponding to the first digit of the number he wishes to contact. His action of releasing the dial is the third interface event in the transaction sequence. This event causes the Transaction state of the OU to change from Committed/Active to Committed/Waiting; and causes the Transaction state of the OU half-system to change from Committed/Waiting to Committed/Active.
4. The local half-system now reads the dialed digit, by rotating the dial from the position set by the user back to the stop. Termination of dial rotation constitutes the fourth event in the transaction sequence.

This event causes the Transaction state of the OU half-system to change from Committed/Active to Committed/Waiting; and causes the Transaction state of the OU to change from Committed/Waiting to Committed/Active.

5. This sequence of transitions continues until the system has read the last dialed digit. Completion of this action causes the Transaction state of the OU half-system to change from Committed/Active to Committed/Waiting; and causes the Transaction state of the non-originating user (NU) half-system to change from Idle/Active to Committed/Active. Both the OU and the OU half-system are now waiting for an event at the NU interface.
6. From the point of view of the model, the entire connection process is internal to the system. As a result, there is no change in model state until the system issues the ringing signal to the non-originating user. This event causes the Transaction state of the NU half-system to change from Committed/Active to Committed/Waiting; and causes the Transaction state of the NU to change from Idle/Active to Committed/Active.
7. The next significant event in the transaction sequence occurs when the called subscriber (non-originating user) answers the call (goes off-hook). This event causes the Transaction state of the NU to change from Committed/Active to Committed/Waiting; and causes the Transaction state of the OU half-system to change from Committed/Waiting to Committed/Active.
8. The OU half-system transfers the non-originating user's answer to the originating user. This event causes the Transaction state of the OU half-system to change from Committed/Active to Committed/Waiting; and causes the Transaction state of the OU to change from Committed/Waiting to Committed/Active.

Strictly speaking, the telecommunication process model cannot be used to represent the transfer of analog voice information, since a user's voice does not constitute a digital source. For the purpose of illustration, we nevertheless portray the user's words as discrete "blocks" of binary information; and imagine that each block is transferred from the source user (speaker) to the telecommunication system (handset microphone) in a discrete event which occurs at a single point in time. We view the transfer of words from the system to the listening user in similar terms, acknowledging the existence of a nonzero time delay between word input and output. To simplify the illustration further, we also assume a unidirectional flow of information from the originating to the non-originating user; and stipulate that the word transfer time is shorter than the time between successive word inputs. Given these assumptions, it is possible to describe the user information transfer process in terms of discrete model events as follows.

9. After receiving the relayed answer signal, the originating user speaks the first word ("block") of the message he wishes to communicate. This event causes three discrete changes in the Output state variable of the originating user: incrementing of the bit and block output indices, and entry of the newly output word into the output record (in sequence, after the last word of the previous transaction). This event also causes the Transaction state of the NU half-system to change from Committed/Waiting to Committed/Active.

10. After a delay corresponding to the word transfer time, a counterpart word is transferred from the NU half-system to the NU. This event causes three discrete changes in the Input state variable of the NU, exactly analogous to the Output variable changes just described. This event also causes the Transaction state of the NU half-system to change from Committed/Active to Committed/Waiting.

11. The sequence of events described in Steps 9 and 10 continues until the OU completes output of all words he wishes to transfer. He then notifies the system of his intent to terminate the transaction by going on-hook. This event causes the Transaction state of the originating user to change from Committed/Active to Closing/Waiting; and causes the Transaction states of the OU and NU half-systems to change from Committed/Waiting to Closing/Active (101).

12. For simplicity, we presume that the NU is notified of the OU's decision to terminate the transaction when he hears the "click" of the latter's on-hook action. Transfer of this "closing signal" across the non-originating user interface causes the Transaction state of the NU half-system to change from Closing/Active to Closing/Waiting (100); and causes the Transaction state of the non-originating user to change from Committed/Waiting to Closing/Active.

13. Within about 110 milliseconds, the switching facilities committed to the transaction at the local central office of the originating user are made available for other transactions; and the line scanner resumes scanning the originating user's line to detect initiation of a new call. In terms of the model, the latter event causes the Transaction state of the OU half-system to change from Closing/Active to Idle/Active; and causes the Transaction state of the OU to change from Closing/Waiting to Idle/Active.

14. The NU responds to the relayed on-hook signal by replacing his own handset on the telephone cradle. This event causes the Transaction state of the NU to change from Closing/Active to Closing/Waiting; and causes the Transaction state of the NU half-system to change from Closing/Waiting to Closing/Active.

15. The NU half-system detects the on-hook signal of the non-originating user; disconnects the NU's line from the trunk which connected it with the originating user; and resumes scanning the line to detect initiation of a new call. The latter event causes the Transaction state of the NU half-system to change from Closing/Active to Idle/Active; and causes the Transaction state of the NU to change from Closing/Waiting to the Idle/Active. This marks the end of the transaction.

There is a difficulty in portraying "resumption of scanning" as a model event in that this event is internal to the system; i.e., in the public switched network there is no signal at the user interface that is coincident with the completion of disconnection. Nevertheless, this event does bear a fixed relationship to the preceeding event (on-hook). In applying the model, we define such events as "referred events", i.e., events whose time of occurrence must be referenced to a previous or subsequent interface signal. Note that the actual delay between the interface event on-hook and the referred event "resumption of scanning" can be measured, by going off-hook at variable intervals after going on-hook and observing whether a dial tone is provided. Such a procedure was suggested by ANSI Task Group X3S3.5 in their character-oriented performance standard X3.44 (ANSI, 1974).

The above event sequence describes a successful information transfer transaction. Obviously, many alternative event sequences can occur when transactions fail. Two such alternatives are briefly described below.

1. System Blocking. In "lost call" switching systems, including the public switched network, the telecommunication system issues a circuit busy (or "system blocking") signal to the originating user when it is unable to establish a circuit path to the intended destination. The originating user must then hang up and try the call again. Such a signal causes the

Transaction state of the OU half-system to change from Committed/Active to Closing/Waiting, and causes the Transaction state of the OU to change from Committed/Waiting to Closing/Active, at step 6 above. The unsuccessful transaction is then terminated in the manner described above, except that the non-originating user is not involved.

2. User Blocking. A transaction will also fail if the non-originating user is busy with another call (i.e., off-hook) at the time the ringing signal would be issued. Such "user blocking" causes the Transaction state of the NU half-system to change from Committed/Active to Closing/Waiting at step 7 above. Transfer of the relayed user blocking signal across the OU interface causes the Transaction state of the OU to change from Committed/Waiting to Closing/Active; and causes the Transaction state of the OU half-system to change from Closing/Active to Closing/Waiting. The call is then terminated in the manner described above, except that again, the non-originating user is not involved.

The public switched network is normally provided to subscribers on a full-time basis. In other types of services, however, a system entity may abruptly end its participation in an information transfer transaction upon expiration of the service time interval. Such an event would be reflected in the model by transition of the terminating entity to the Idle/Waiting state. Various other abnormal events of lesser interest can also be represented in terms of model state transitions.

2.5.2. Message Telegram Transaction

The process of sending a telegram via public message telegram service can be summarized very briefly as follows.

1. The user wishing to send the telegram (source user) places a conventional voice telephone call over the public switched network to the local telegraph switching office, and dictates his message orally to

the telegraph operator. After the operator has copied the complete message, both he and the source user hang up.

2. The local telegraph operator transfers the source user's message to his counterpart at the local telegraph office closest to the intended destination user, typically employing the store-and-forward switching and both voice and digital (e.g., TELEX) transmission. Neither the source user nor the destination user is involved in this process, and either or both of these entities can be involved in other transactions during this process.
3. After receiving the complete message (typically, typed at a TELEX terminal) the destination user's local telegraph operator places a conventional voice telephone call over the public switched telephone network to the destination user; and reads him the typed message. The operator and the destination user then hang up, completing the end-to-end transaction. (We disregard the "hand delivery" and "mailgram" options here.)

In considering this transaction from the point of view of the telecommunication process model, a number of points are apparent:

1. The telephone call between the source user and his local telegraph operator involves exactly the same sequence of interactions as the user-to-user voice telephone call illustrated in Figure 2.9, except that there is only one end user interface, and no destination user is involved. The event sequence of Figure 2.9 can be used to represent this phase of the telegram transaction by simply placing the non-originating (destination) entity within the system (which is, of course, exactly the position of the source telegraph operator). From the end user point of view, then, only the events indicated at the OU interface occur during this phase of the end-to-end transaction.

2. The interactions by which the message is transferred between the source and destination telegraph operators are wholly internal to the system, and hence involve no user interface events or model state transitions.
3. The telephone call between the destination telegraph operator and the destination user is also exactly analogous to the user-to-user call illustrated in Figure 2.9, except that again, there is only one user interface: this time, no source user is involved. The event sequence represented in Figure 2.9 can be used to represent this phase of the transaction by simply placing the originating (source) entity within the system (which is, again, exactly the position of the destination telegraph operator). From the end user point of view, only the events indicated at the destination user interface occur during this phase of the end-to-end transaction.

Thus, in the case of the message telegram transaction, the sequence of events occurring at each user interface (viewed separately) is identical to the corresponding sequence in the user-to-user voice telephone transaction; but the time relationship between the two event sequences is totally different. In the circuit switched case, the two event sequences are overlapped in time; in the message switching case, the two event sequences are separated in time by a period of minutes or even hours.

Our objective in describing these familiar transaction profiles has been to illustrate the general operation of the telecommunication process model in readily understandable terms. Although these examples are, of course, not directly representative of data communication transactions, there are message-switched and circuit-switched digital services with user/system interaction sequences very similar to those described. Packet switching services normally employed either a "virtual circuit" protocol, which is directly analgous to the circuit switched example; or a "datagram" protocol, which is directly analogous to the message switched example. The ability of the model to represent these diverse protocols provides some evidence of its broad applicability. More specific examples of its application to digital services are presented in Volume II.

3. PRIMARY PARAMETERS

3.1. Introduction

This section summarizes the technical considerations which influenced selection of primary performance measures for the proposed Federal Standard; and defines the primary functions, outcomes, and parameters on the basis of the telecommunication process model of Section 2. The section is divided into three major subsections, corresponding to the Function Definition, Failure Analysis, and Parameter Selection steps in the parameter development process.

3.2 Function Definition

Performance can only be considered in the context of a particular well-defined function. Section 2 pointed out that any ordered set of model state transitions can be used to define a communication function. This subsection describes the process by which this very large set of possible functions was reduced to the five primary functions included in the proposed standard; and defines these selected functions in terms of user/system interface signals and associated model state transitions.

3.2.1. Technical Considerations

The primary functions were selected on the basis of the four desired parameter attributes listed in Section 1.1: user orientation, universal applicability, simplicity, and completeness. The objective of user orientation imposes two requirements: (1) that selected parameters be defined in terms of events directly observable to end users; and (2) that parameters be chosen on the basis of expressed user performance concerns. The first requirement was essentially satisfied in the design of the telecommunication process model. The model allows only two measurement interfaces, one associated with each end user; and describes the results of all communication activities in terms of discrete interface events. The parameters thus can only be defined in terms of user-observable events.

The Subcommittee addressed the second requirement by surveying various available sources of user input, including (1) the industry responses to the FCC's Quality and Reliability Inquiry (Docket 18920) and earlier Computer Inquiry (Docket 16979); (2) comments of the Federal government user agencies represented on FTSC; (3) earlier published performance standards (e.g., ANSI, 1974); and (4) technical literature on teleprocessing and data communication performance assessment. This survey indicated that users have distinct performance concerns associated with each of three basic types of communication activities:

1. Access Activities. These activities place the system and the users in a position to begin transferring user information.
2. User Information Transfer Activities. These activities effect actual transfer of user information between and across the user/system interfaces.
3. Disengagement Activities. These activities terminate the conditions that enabled user information transfer between specified users. They release allocated system facilities for subsequent use and enable the users to participate in subsequent transactions.

Access, user information transfer, and disengagement were therefore chosen as the three major functional categories to be considered in defining the primary functions. It was anticipated that one or more specific functions would be chosen to describe performance within each category.

The objective of universal applicability requires that all selected parameters be capable of describing the performance of any end-to-end telecommunication service, independent of network unique characteristics such as topology and control protocol. This objective was a primary consideration in the design of the telecommunication process model. But universality of the model is not sufficient to insure universal applicability of the performance parameters, since functions defined on a universal model can still be system dependent. As an example, we could define a function on the basis of the Transaction state variables of the originating user and half-system as follows:

Function Delimiter	Transaction State Transition	
	OU	HS
Starting Event	001→010	001→011
Ending Event	010→011	011→010

In the public switched network, the starting event above corresponds to going off-hook; the ending event corresponds to the issuance of dial tone; and the function is to provide dial tone. "Dial tone delay" is a commonly used parameter whose definition is based on this function.

The function of providing dial tone clearly is not universal, since its defining events are not essential to completion of an information transfer transaction in every system. As an example, in the ARPA network, the equivalent of off-hook is transfer of a Connect request from an applications program (originating user) to a network control program (half-system) within a Host computer. The Connect request specifies the equivalent of the non-originating user's telephone number (a receive socket number) by reference to a stored Connection Control Table. Once the Connect request has been issued, the network control program can read the stored socket number unilaterally, without transferring responsibility back to the originating user with the equivalent of a dial tone signal. The function of providing dial tone and the parameter "dial tone delay" thus cannot be defined in the ARPA network.

The objective of universal applicability can only be achieved if the primary functions are defined on the basis of events that must occur, in any system, if the transfer of user information between end users is to be effected. One such event is the start of bit transfer: if a bit of user information is to be moved between end users, it must at some point move from the physical possession and control of the source user to that of the system. The identification of such universal reference events was a major concern of the Subcommittee.

The third and fourth desired attributes, simplicity and completeness, were viewed as essentially opposing requirements to be "traded off" during the parameter development process. The objective of simplicity requires that the selected functions be few in number, simply defined, and readily measured. The objective of completeness requires that the selected functions not be so few or so elementary that they omit performance factors of interest to users, or mask significant performance differences between functionally equivalent services. During the development of the standard, these general objectives were translated into three more specific requirements:

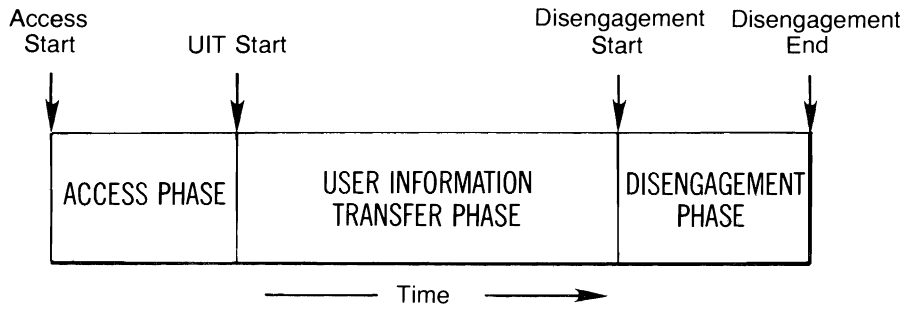
1. No function would be included in the standard unless it could be clearly correlated with a specific user performance description need, as expressed in the surveyed sources.
2. To avoid defining complex functions aggregating fundamentally different types of communication activities, each selected function would be restricted to one of the three classes of activities described above; i.e., access, user information transfer, or disengagement.
3. To encourage completeness, the functions would be defined such that all periods of performance within

an information transfer transaction would be covered by some function (i.e., no "gaps").

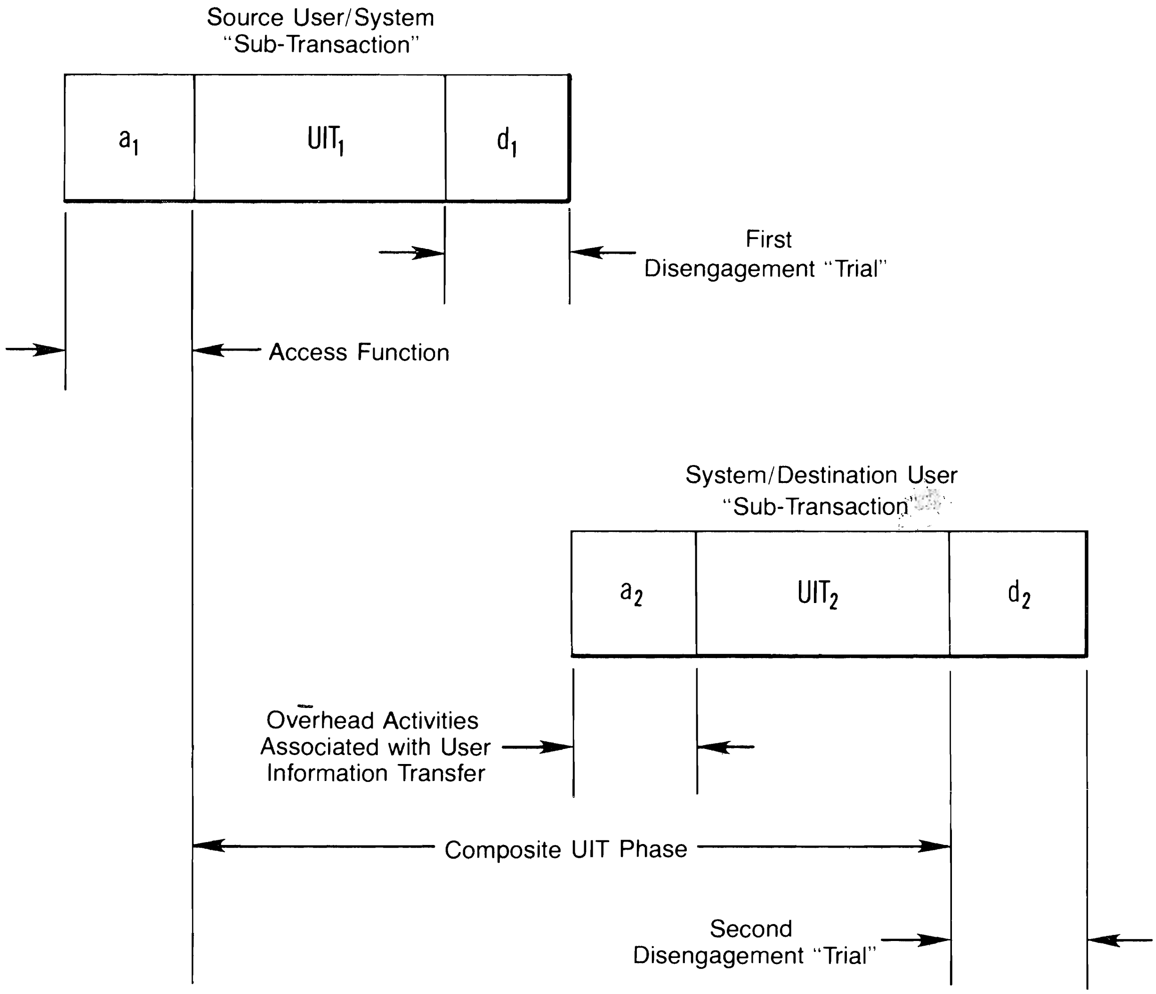
Three key technical issues also influenced the selection of primary functions. The first was the issue of functional equivalence - how to define the primary functions in such a way that the same performance parameters could be applied to both circuit-oriented and message-oriented services. Figure 3.1 illustrates the essential difference between these two types of service from a performance assessment standpoint. In a circuit-oriented transaction (Fig. 3.1a), the access, user information transfer, and disengagement activities are essentially concurrent at the two user interfaces; and the overall transaction can be neatly divided into three consecutive, non-overlapping functional phases: access, user information transfer, and disengagement. For systems employing circuit-switched or virtual circuit protocols, it would therefore be logical to describe performance in terms of:

1. A single access function, encompassing activities between the start of the transaction and the start of actual user information transfer;
2. One or more user information transfer functions, encompassing activities between the end of access and the start of disengagement; and
3. A single disengagement function, encompassing activities between the end of user information transfer and the end of the overall transaction.

There is a problem in applying this approach directly to message-oriented services, as Figure 3.1b illustrates. Here, the end-to-end transaction is comprised of two "sub-transactions" which are independent and disjoint in time: one between the originating user and the telecommunication system, and one between the system and the non-originating user. Each sub-transaction could be regarded as having a distinct access, user information transfer, and disengagement phase, analogous to the phases defined in the circuit switched case; but it is not



a. Circuit-Oriented Transaction



b. Message-Oriented Transaction

Figure 3.1. Circuit-oriented vs. message-oriented transactions.

clear how to relate these message-oriented phases to their circuit-oriented equivalents.

This general issue was divided into three separate problems, corresponding to the three activity phases; and each problem was treated in a slightly different way. The user information transfer (UIT) activities represent the most straightforward case. Clearly, from the end user point of view, the transfer of a unit of user information begins when the first bit of the information unit crosses the source user/system functional interface, and ends when the last bit of the information unit crosses the system/destination user interface, irrespective of whether circuit switching or message switching is employed. Equivalent UIT functions can thus be assured by defining each function in terms of corresponding source user output and destination user input events. User information transfer activities extend over the entire "composite UIT phase" indicated in Figure 3.1b in the message-oriented case; but not all activities within this period are associated with user information transfer.

It is slightly more difficult to establish functional equivalence between the disengagement activities in message-oriented and circuit-oriented services. In the message-oriented case, there clearly are two independent functions being performed: disengagement of the originating user from the system, and disengagement of the non-originating user from the system. In general, the success or failure of one function has no impact on the outcome of the other; and either function can fail without influencing the access or user information transfer activities during that particular transaction.

In the circuit-oriented case, it is not as clear that disengagement of the originating user and disengagement of the non-originating user are independent functions: the disengagement activities at two interfaces take place concurrently, and thus can either be associated or separated functionally depending on the detailed selection of starting and ending events. It is true, however, that a circuit-switched network can fail to disengage one of the two interconnected users while successfully disengaging the other.

Three approaches to the definition of equivalent disengagement functions were considered:

1. Express disengagement performance in terms of one function - the disengagement of both users. This option favors the conceptually simpler circuit-oriented case.
2. Express disengagement performance in terms of two separate functions - originating user disengagement and non-originating user disengagement. This option favors the message-oriented case, and provides a more detailed description of overall performance, at a cost of increased specification and measurement effort.
3. Express disengagement performance in terms of one function - the disengagement of either user. Under this option, originating user disengagement (d_1) and non-originating user disengagement (d_2) would be regarded as separate "trials" of a general disengagement function; and a normal information transfer transaction would include two disengagement "trials." This option obscures possible differences between the disengagement performance observed by the originating and non-originating users; but provides the advantage of specification and measurement simplicity.

The latter approach was ultimately selected by the Subcommittee; and accordingly, disengagement performance is described (in both circuit-oriented and message-oriented services) in terms of a single primary function, simply termed disengagement. The function encompasses the activities required to disengage a single committed user, who may represent either the originating or the non-originating user.

The same considerations described above also apply to the definition of equivalent access functions. Here, however, there is one significant difference. In message-oriented services, the interface events associated with non-originating user commitment occur after the start of user information transfer, and are essential to its successful completion. If a "non-originating user access" function (a_2) were defined to represent these activities, it would overlap, and be redundant with, the user information transfer functions. It was therefore concluded that in describing both message-oriented and circuit-oriented services, only activities occurring prior to the start of user information transfer would be regarded as access activities; and a single access function would be defined to represent them.

The above approach achieves the goal of functional equivalence only with some sacrifice in descriptive precision. Under the chosen definitions, the activities which produce non-originating user commitment are associated with the access function in circuit-oriented systems, but are associated with the user information transfer functions in message-oriented systems. The two classes of systems can nevertheless be represented by the same outcome sample spaces and performance parameters, as described in Section 3.3.

The chosen approach does have the merit of being consistent with intuitive expectations about the relative performance of circuit-switched and message-switched systems. For example:

1. Access times tend to be longer in circuit-switched systems, because the non-originating user must be contacted and "committed" to the transaction prior to the start of user information transfer.
2. Message transfer times tend to be longer in message-switched systems, in part because this same function of contacting and committing the non-originating user must be performed after the start of user information transfer. Message-switched systems must store the user information accepted from a source until the destination is committed to receive it; and the storage process itself adds to the end-to-end message transfer time.

3. Data loss probabilities tend to be lower in circuit-switched systems than in message-switched systems. In the former case, a fixed "real time" circuit is established between end users (and in many cases, is confirmed by them) prior to the start of user information transfer. In the latter case, the source user must commit his message to the system without confirming that the intended destination is ready to receive it.

A second key issue which was addressed in defining the primary functions was the "level of detail" which should be used in describing user information transfer. Various user information units, and corresponding user information transfer functions, can be considered: bit transfer, block transfer, transfer of a defined number of bits, a defined number of blocks, and so on. The issue then, was to select an information unit (or group of information units) to be used as the basis for describing user information transfer performance.

Each of the four desired parameter attributes had a significant effect on the resolution of this issue. A primary reason for seeking universal applicability in the selected parameters was to enable comparison of alternative services in terms of a universal "common denominator". The block lengths used in transmitting user information vary widely between systems; and it is neither desirable nor feasible to mandate a "standard block length" in order to facilitate performance measurement. Nevertheless, alternative systems can always be compared at the bit level; and it was therefore concluded that the user information transfer functions should include a primary function of bit transfer.

The term "block" is generally defined as a contiguous group of bits delimited for transfer or processing as a composite whole. For our purposes, we further restrict the term block to an information unit delimited at the user/system interface, consisting solely of user information bits, i.e., a <u>user</u>

information block. Examples of typical user information blocks are the five bits in a Baudot character (generated by the operator of a character-asynchronous data terminal); the 960 bits in a standard 12x80 punched card (generated by a programmer for input to a remote computer); and an individual record in a binary output file (generated by an applications program in an automatic teleprocessing network). A block as defined herein does not correspond to a data link control or error control "frame" of the type defined in (say) the ANSI Advanced Data Communication Control Procedure (ANSI, 1975); such frames contain overhead information in addition to user information, and are delimited inside the user/system interface. The reason for excluding overhead bits in defining user information blocks is that overhead bits normally do not cross both end user interfaces, and hence cannot be used in defining end-to-end telecommunication functions.

The user information unit termed a "block" clearly cannot be used as a standard unit in comparing alternative systems. Nevertheless, the block is very often the most significant information unit for the end user. In character-oriented systems, for example, bits are simply a convenient way of representing characters (blocks) for transmission within the system; the significant information unit is the character. It was therefore concluded, on the basis of user orientation and completeness, that the UIT functions should include a primary function of block transfer. In the interest of simplicity, it was decided that the block-oriented parameters should be based on a single (average) block length, to be specified by the telecommunication system operator.

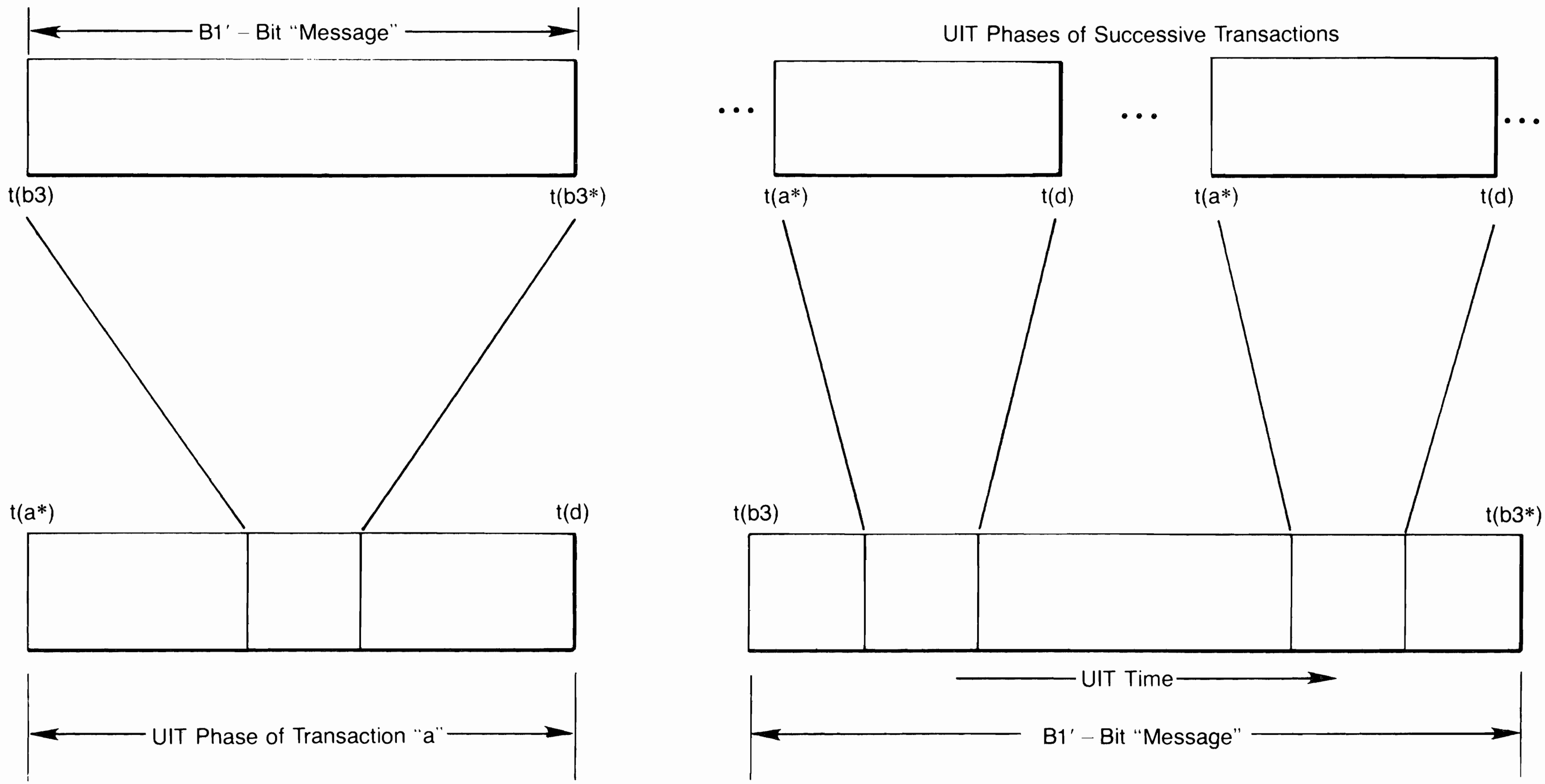
It was the view of the Subcommittee that a third primary UIT function should be specified, both to permit definition of longer-term UIT parameters such as rates, and to provide an appropriate sample size for calculating bit transfer and block transfer parameter values. Three alternative ways of defining a suitable multi-block information unit (or "message") were considered:

1. The quantity of bits transferred during a single information transfer transaction.
2. A fixed number of blocks, sufficient to produce a given level of confidence in the measured values for selected block-oriented parameters.
3. A fixed number of bits, sufficient to produce a given level of confidence in the measured values for selected bit-oriented parameters.

The first alternative is appealing from the standpoint of simplicity; but the length of an information transfer transaction is highly variable under typical operational conditions. Measured parameter values based on this information unit would not be comparable, either among users or between successive transactions involving the same user. This alternative was therefore discarded as inconsistent with the objective of universal applicability. The alternative of requiring a fixed information transfer transaction length for the purpose of measuring performance was judged to be unacceptable to most users.

The choice between the latter two options was made on the same general basis. It was anticipated that primary parameter values calculated over the selected message length would be used to identify system outages, and to define the secondary (availability) parameters. Values for the block-oriented parameters could not be used in this way since they would not be comparable between services having different block lengths. The selected message length was therefore specified as a fixed number (Bl') of bits. The prime mark (') indicates that the sample excludes bits not delivered due to user nonperformance, as described in Section 3.4.2. Methods of relating the number Bl' to required confidence limits will be specified in the measurement standard.

Figure 3.2 illustrates the consequences of the decision to describe user information transfer performance in terms of a "message" length which may differ from the number of bits trans-



a. "Message" Length within Single Transaction

b. "Message" Length Covers More than One Transaction

Figure 3.2. Alternative formulations of the message transfer function.

ferred during an information transfer transaction. There are two situations of interest:

1. The chosen message length may be contained within a single information transfer transaction (Fig. 3.2a).
2. The chosen message length may cover more than one transaction (Fig. 3.2b).

The first situation poses no particular problem: the message transfer performance time is continuous in clock time, since it is contained within the longer user information transfer performance period. The second situation is somewhat more complex. Here, the period of user information transfer during an individual transaction is not long enough to produce a sample population of Bl' transferred bits; consequently, two or more information transfer transactions must be linked together in the performance record to define a message transfer function. In this case, the message transfer performance time is discontinuous in clock time, and is comprised of a succession of user information transfer performance periods, excluding all time outside the user information transfer phase. The discontinuous time scale defined in this way is termed user information transfer time (UIT time).

The third key issue addressed by the Subcommittee in selecting primary functions was the issue of user dependence. Two basic types of functions can be defined on the telecommunication process model of Section 2:

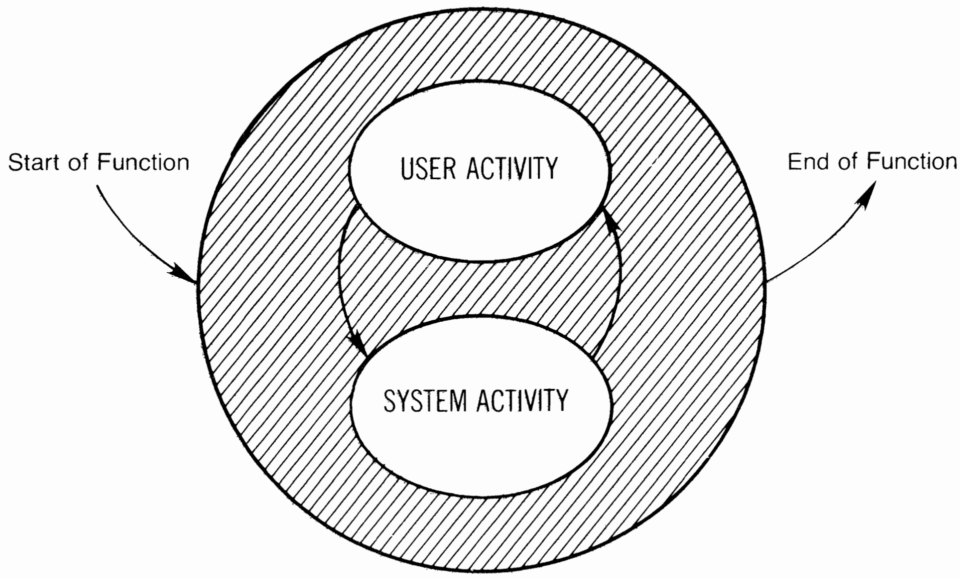
1. Functions that include no intermediate events which must be produced by a user. Since these functions describe periods of unilateral system activity, performance parameters defined on them are independent of user influence.
2. Functions that include one or more intermediate events which must be produced by a user. Such functions involve user activity as well as system activity; and any parameters defined on them will describe the joint performance of both entities.

A basic issue which was discussed at length in the Subcommittee was whether functions of the second type, i.e., user dependent functions, should be allowed in the proposed Federal Standard. There is an obvious problem in using parameters defined on such functions to specify required system performance: the carrier or other organization responsible for supplying the telecommunication service normally has no control over user performance, and hence cannot ensure that user dependent parameter values will be met. Nevertheless, many of the parameters which best describe end-to-end telecommunication performance are based on user dependent functions. A familiar example is "throughput," which depends (in general) on the rate at which the user inputs information to the telecommunication system.

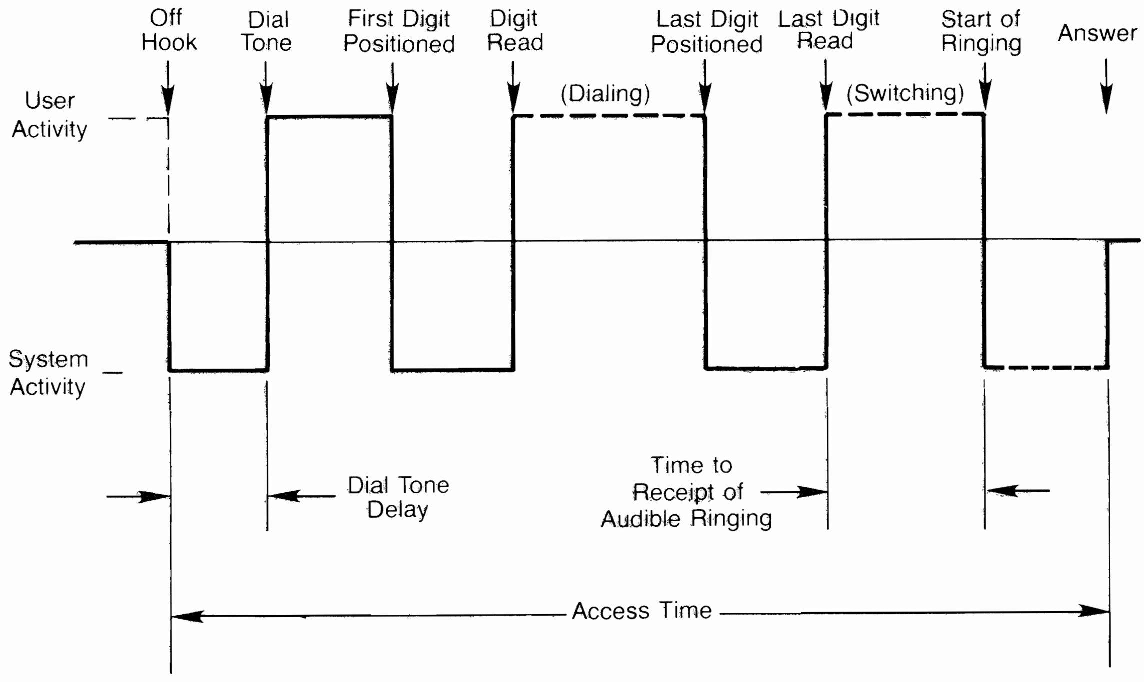
Three general approaches to the user dependence issue can be considered (Fig. 3.3). One obvious approach is to avoid the problem altogether by excluding user dependent functions. The operation of a system can always be described in terms of a succession of stimulus/response functions, each requiring only a single (initial) user input. This approach has traditionally been employed by the common carriers in describing communication system performance. As an example, one parameter commonly used to describe access performance in the public switched network is the "time to receipt of audible ringing" - the elapsed time from the end of the user's dialing to the start of ringing. This parameter is defined to exclude the user functions of dialing and answering; and it therefore provides a measure of "pure" system performance which is independent of user behavior. Another parameter in the same category is "dial tone delay."

The difficulty with the above approach is that it leads directly to performance parameters which are system dependent. Such parameters cannot be used to compare the performance of different systems, as was described above.

A second approach to the user dependence problem is to define the functions of interest as a concatenation of disjoint system activities, excluding intervening user activities. To continue



a. Alternating Activities within a User Dependent Function



b. Voice Telephone Access Illustration

Figure 3.3. User dependence in the measurement of access performance.

the telephone access illustration, we could, for example, define a system function in establishing voice contact between two users to include (1) providing dial tone in response to an off-hook (input) signal; (2) reading each dialed digit (by rotation of the dial back to the stop); and (3) interconnecting the calling and called parties' telephones, as evidenced by the start of ringing or called party busy signal. The composite of these separate activities could be regarded as a single, user independent function.

This second approach also has a major disadvantage from the user point of view: it does not reflect differences in the "functional burden" placed on the user by different systems providing the same ultimate service. Referring again to the voice telephone access illustration, this second approach would not reflect (in terms of "better" parameter values) the obvious advantage of push button dialing over traditional rotary dialing.

The third possible approach to the problem of user dependence is to allow definition of user dependent functions, but describe the user component of aggregate system/user performance in terms of separate "ancillary" parameters. The Subcommittee concluded that the latter approach provided the best compromise from the point of view of the desired parameter attributes. A detailed description of this approach is presented in Section 5.

3.2.2. Primary Communication Functions

The above technical considerations led to the selection of five primary communication functions: access, bit transfer, block transfer, message transfer, and disengagement. Table 3.1 defines the selected functions, in terms of overhead and user information transfers and associated model state transitions.

> The access function begins on issuance of an "access request" signal at the originating user interface; and ends (successfully) on transfer of the first user information bit from a source user to the system. The starting event is evidenced by transition of the Transaction state of the originating user and his local half-system from Idle/Active

Table 3.1. Primary Functions

FUNCTION INDEX	FUNCTION NAME	STARTING EVENT		ENDING EVENT	
		TRANSFERRED INFORMATION	STATE TRANSITION	TRANSFERRED INFORMATION	STATE TRANSITION
a	ACCESS	ACCESS REQUEST	ENTITY TS OU: 001→01X HS: 001→01X	FIRST USER INFORMATION BIT	SU OUTPUT RECORD: b1 =b1 +1
b1	BIT TRANSFER	USER INFORMATION BIT b1	SU OUTPUT RECORD: b1 =b1 +1	USER INFORMATION BIT b1	DU INPUT RECORD: b1 =b1 +1
b2	BLOCK TRANSFER	USER INFORMATION BLOCK b2	SU OUTPUT RECORD: b2 =b2 +1	USER INFORMATION BLOCK b2	DU INPUT RECORD: b2 =b2 +1
b3	MESSAGE TRANSFER	USER INFORMATION MESSAGE b3	SU OUTPUT RECORD: b3 =b3 +1	USER INFORMATION MESSAGE b3	DU INPUT RECORD: b3 =b3 +1
d	DISENGAGEMENT	DISENGAGEMENT REQUEST	ENTITY TS U OR HS: X1X→X0X	DISENGAGEMENT CONFIRMATION	ENTITY TS U: X0X→00X HS: X0X→00X

Legend

OU = Originating User
HS = Half-System
SU = Source User
U = Either (Specified) User
TS = Transaction State
DU = Destination User

Note

In the case where a transferred information unit crosses the user/system functional interface in a series of increments, the starting event occurs when the first increment is transferred, and the ending event occurs when the last increment is transferred.

Transaction States

XXX

Closing State (first X)
0 = Not Closing
1 = Closing

Commitment State (second X)
0 = Uncommitted
1 = Committed

Responsibility State (third X)
0 = Entity Waiting
1 = Entity Active
X = Either State

to Committed ($001 \to 01X$). The ending event is evidenced by the first subsequent incrementing of the bit transfer index in the Output state variable of the source user (b1=b1+1).

The bit transfer function begins on transfer of a user information bit from the source user to the system; and ends (succcessfully) on transfer of the corresponding bit from the system to the destination user. Each interface event is evidenced by the incrementing of a bit index in the appropriate (Output or Input) state variable (b1=b1+1).

The block transfer function begins on start of transfer of a user information block from the source user to the system; and ends (successfully) on completion of transfer of the corresponding block from the system to the destination user. Each interface event is evidenced by the incrementing of a block index in the appropriate (Output or Input) state variable (b2=b2+1).

The message transfer function begins on start of transfer of a user information message from the source user to the system; and ends (successfully) on completion of transfer of the corresponding message from the system to the destination user. Each interface event is evidenced by the incrementing of a message index in the appropriate (Output or Input) state variable (b3=b3+1).

The disengagement function begins on issuance of a "disengagement request" signal relative to either user interface. The end of the function is either coincident with, or is referred to, issuance of a subsequent "disengagement confirmation" signal relative to the same user interface. The starting event is evidenced by transition of the user or half-system at the specified interface out of the Committed Transaction state ($X1X \to X0X$); the ending event is evidenced by transition of both entities into the Idle state ($X0X \to 00X$).

The terms "access request," "disengagement request," and "disengagement confirmation" are general descriptors of function rather than specific names of interface signals. They denote, respectively, any signal whose purpose is to initiate, terminate, or confirm termination of an entity's participation in an information transfer transaction. Typical access request signals are the Connect request in the ARPA network; the Open Destination (OPNDST) VTAM macro in IBM's SNA network; and the off-hook signal in manual teletypewriter services using the public switched network. Corresponding disengagement request signals are the Close request; the Close Destination (CLSDST) macro; and the "on-hook" signal. In the first two examples, the end of disengagement is indicated by an explicit disengagement confirmation signal (Connection Control Table and Event Control Block entries, respectively). In the latter example, the end of disengagement is an implied event, whose time of occurrence must be estimated using the procedure described in Section 2.5.1. A separate disengagement function is defined for each user interface. A user-initiated disengagement request can refer either to the local interface only, or to both user interfaces.

In the case where transferred signals cross the user/system interface in a series of increments, starting events are defined to occur when the first increment is transferred; and ending events are defined to occur when the last increment is transferred. Information is said to have been transferred between functional entities (or to have "crossed the functional interface" between a sending and receiving entity) when the following two conditions have been met:

1. The information is physically present within the facilities used to support the receiving entity.
2. The receiving entity has been notified, or is aware, that it can begin activities involving use of that information.

3.3. Failure Analysis

The preceding subsection described "what should happen" during a "successful" information transfer transaction in terms of five primary functions. This subsection describes "what could happen" during such a transaction, by defining a discrete "sample space" of possible outcomes for each primary function. A systematic enumeration of possible performance failures is obviously an essential step in selecting performance parameters.

3.3.1. Sample Space Concepts

In the following paragraphs, we will represent the possible outcomes of a trial performance of each primary function in a discrete probability sample space; and will use these sample spaces as the basis for defining the primary performance parameters. This paragraph provides a brief summary of discrete sample space concepts which are essential to the outcome and parameter definitions. The paragraph is tutorial in nature and can be omitted by readers familiar with elementary probability theory.

A sample space G associated with a real or conceptual experiment is a set of elements $\{g_i\}$ such that (1) each element g_i denotes an outcome of the experiment, and (2) any performance of the experiment results in an outcome that corresponds to one and only one element of G (Goldberg, 1960). A sample space formally defines an experiment, and serves as a universal point of reference for all questions concerning the experiment.

The individual elements of a sample space are termed elementary events or sample points. We will use the latter term to avoid confusion with the term "event" as it is used in the modeling context (Sec. 2). Logical unions of sample points are conventionally termed composite events. We will refer to composite events as subsets of the sample space for the same reason.

Each sample point g_i in a discrete sample space of m elements can (in concept) be assigned a probability $P(g_i)$, such that

$$0 \le P(g_i) \le 1$$

and

$$\sum_{i=1}^{m} P(g_i) = 1.$$

This so-called "acceptable assignment" of probabilities to the sample points provides a basis for predicting or summarizing experimental outcomes in numerical form.

A simple illustration of the sample space concept is the experiment of tossing a die. This experiment can be represented by the set of sample points $\{g_i | 1 \leq i \leq 6\}$, each point corresponding to one of the six numbered faces. A probability $P(g_i)=1/6$ would normally be assigned to each outcome. The same experiment can also be represented by various less detailed sample spaces, e.g. {even, odd} or $\{6,\overline{6}\}$, if only this more limited information is of interest. In general, a sample space should distinguish only those outcomes of separate interest to the experimenter.

One additional concept used in defining the communication performance outcomes is the concept of a product or composite sample space. Given the die experiment described above, we may wish to consider the possible outcomes of the composite experiment of rolling the die twice (or "n" times) and observing the result on each roll. These outcomes can be represented in a single composite sample space which is the Cartesian product (or "cross product") of the individual spaces G. In the case where n=2, the composite sample space contains 36 sample points, each denoting a particular pair of single-toss outcomes. In the general experiment of n trials, the sample space contains 6^n points, each point comprising a vector of n single-toss outcomes.

Sample spaces provide a very useful basis for defining performance outcomes and parameters. They encourage the analyst to consider all possible outcomes of a performance trial; to specify precisely the criteria that distinguish each outcome; and to evaluate candidate parameters systematically, on the basis of stated performance description needs. Properly defined sample spaces also ensure that measured probability parameter values will be "well behaved", i.e., will remain within the range $0 \leq P \leq 1$, under all possible conditions of performance.

3.3.2. Overall Approach

This paragraph defines a discrete outcome sample space for each of the five selected primary functions. To establish a common basis for these definitions, we first consider the possible outcomes of a general function g, which may represent any one of the five. The function is a defined sequence of user/system interface events, delimited by specified starting and ending events and including, in general, one or more intermediate events (Fig. 3.4a). Each interface event consists of the transfer of a discrete unit of overhead or user information between the indicated user/system entities.

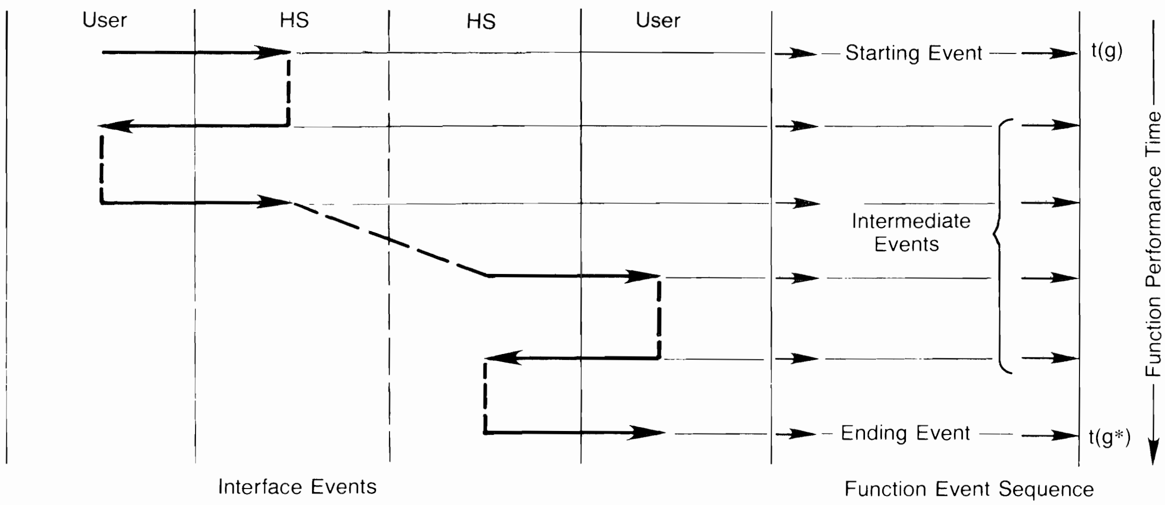
Two general types of performance observations can be made during the performance of such a function: observations of interface events, and separately, observations of the time of occurrence of these events.

The possible interface events can be divided into three general categories (Fig. 3.4b):

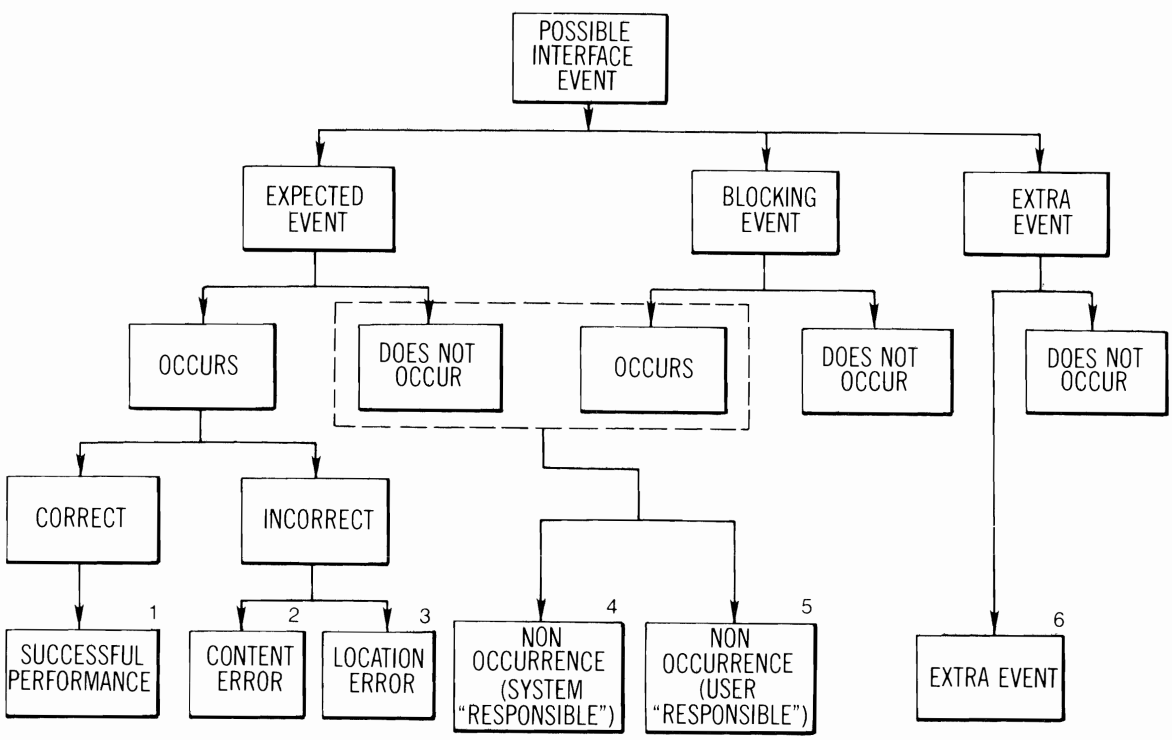
1. Expected Events. This category includes all events which necessarily occur during successful performance of the function.
2. Blocking Events. This category includes all events whose occurrence signifies that the function in progress cannot be successfully completed. Examples are circuit busy and user busy signals in "lost call" systems.
3. Extra Events. This category includes all other events. Examples of extra events are delivery of unwanted bits in a user information block; and delivery of "duplicate blocks" within a multi-block message.

Observation of performance relative to an expected event will result in one of four outcomes:

1. The event occurs as expected. The correct information is transferred across the intended user interface within the specified maximum function performance time.
2. The event occurs at the intended user interface, but with one or more errors in the transferred information content.



a. Event Sequence for a Typical Primary Function, g



b. Event-Related Observations

Figure 3.4. Event outcomes.

3. The event occurs at an unintended user interface (i.e., at the system interface with a user not intended to be involved in the transaction).

4. The event fails to occur at all within the specified maximum function performance time.

To address the issue of user dependence, the expected event nonoccurrences must be divided into two classes, based on whether the user or the system is "responsible." The extra event occurrences must also be added, as a separate outcome class. We thus have six possible outcomes of the conceptual experiment of making a single event-related performance observation relative to a primary function:

1. Successful Performance. The expected event occurs, and is correct in content and location.

2. Content Error. The expected event occurs at the correct location, but is incorrect in content.

3. Location Error. The expected event occurs at an incorrect location.

4. System Nonperformance. The expected event does not occur within the maximum performance period, as a result of either issuance of a blocking signal or excessive delay on the part of the telecommunication system.

5. User Nonperformance. The expected event does not occur within the maximum performance period, as a result of either issuance of a blocking signal or excessive delay on the part of the users.

6. Extra Event. A nonblocking event, not included within the expected event sequence, occurs.

These outcomes are represented graphically in the "pie" diagram shown in Figure 3.5. The outcomes are grouped into three categories to indicate the three performance concerns most frequently expressed by users: efficiency (or "speed"), associated with instances of successful performance; accuracy, associated with instances of incorrect performance; and reliability, associated with instances of nonperformance. These three performance

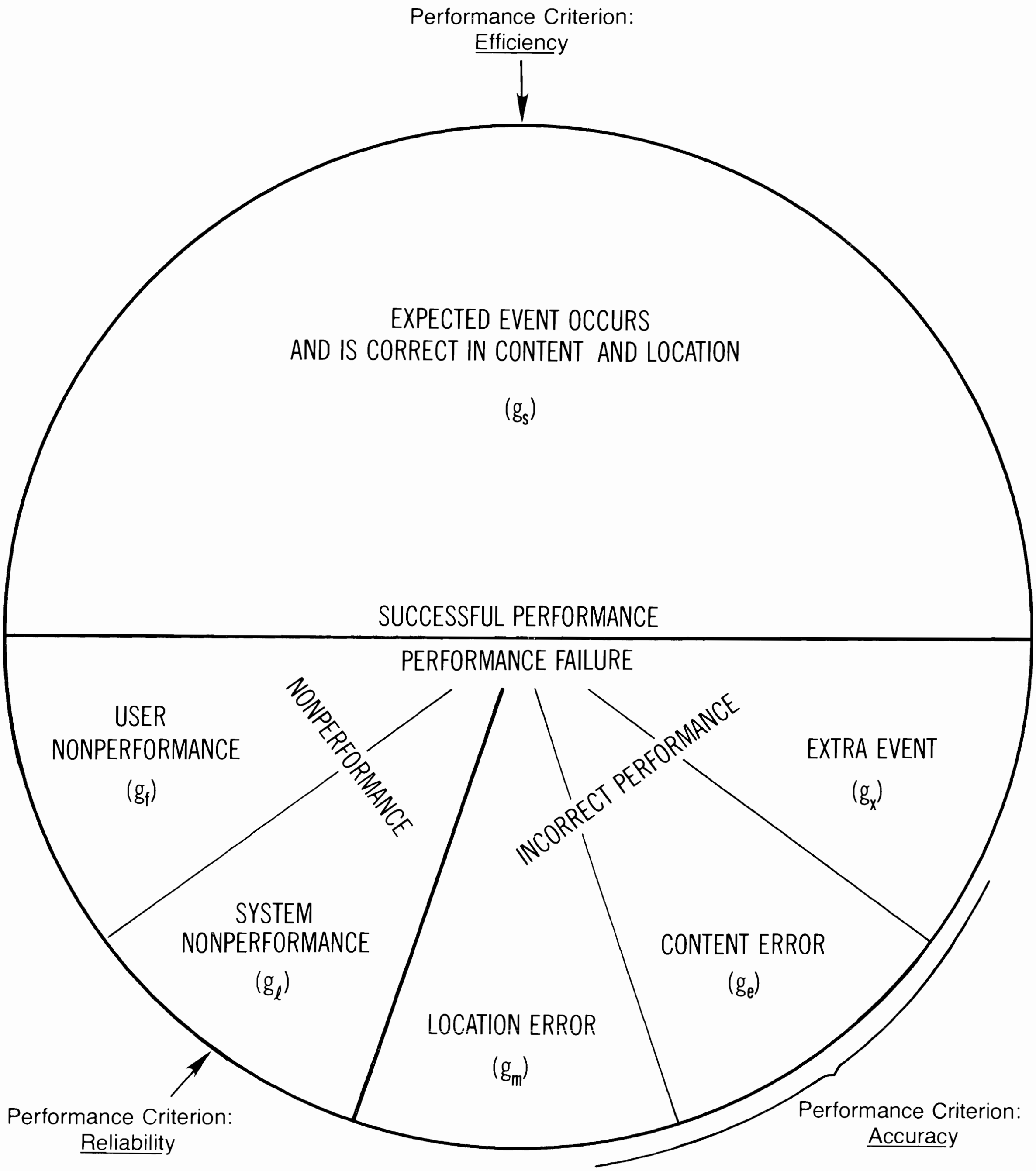


Figure 3.5. Sample space for a single event observation.

criteria provided a useful general structure within which all candidate performance parameters were considered. The sample space of Figure 3.5 can also be represented in symbolic form, $G=\{g_i | 1 \le i \le 6\}$, by simply numbering the event outcomes. This form of representation is used in defining performance parameter options in Section 3.4.

Of the five selected primary functions, four can be represented either directly by the event sample space of Figure 3.5, or by an appropriate subset of it: access, bit transfer, block transfer, and disengagement. The bit transfer and disengagement functions are defined such that their outcomes can be determined by a single event observation; and the appropriate single-event outcomes of Figure 3.5 thus apply directly to each function as a whole. The access and block transfer functions involve (in general) multiple event observations; and we would therefore expect their outcome sample spaces to have the "vector" form described in Section 3.3.1. It is nevertheless possible to represent the possible outcomes of these functions in terms of the sample space of Figure 3.5, by combining outcomes that need not be distinguished from the end user point of view.

Figure 3.6 shows the general form of the event sample spaces which represent the message transfer function. Two separate sample spaces are defined to represent this function: a bit-oriented sample space and a block-oriented sample space. Each sample space is a simplification of the general "vector" space described in Section 3.3.1, produced by combining all composite outcomes having the same ultimate <u>distribution</u> of individual bit or block outcomes among the six outcome categories, irrespective of their order of occurrence. The "heights" of the outcome "slices" in Figure 3.6 indicate the total numbers of bit transfer and block transfer trials encountering each outcome. Each sample point g_i in the composite space can thus be viewed as a set of six "occupancy numbers," $\{G_1 G_2 \ldots G_6\}$, each number representing the total number of bit (or block) transfer trials encountering a particular outcome.

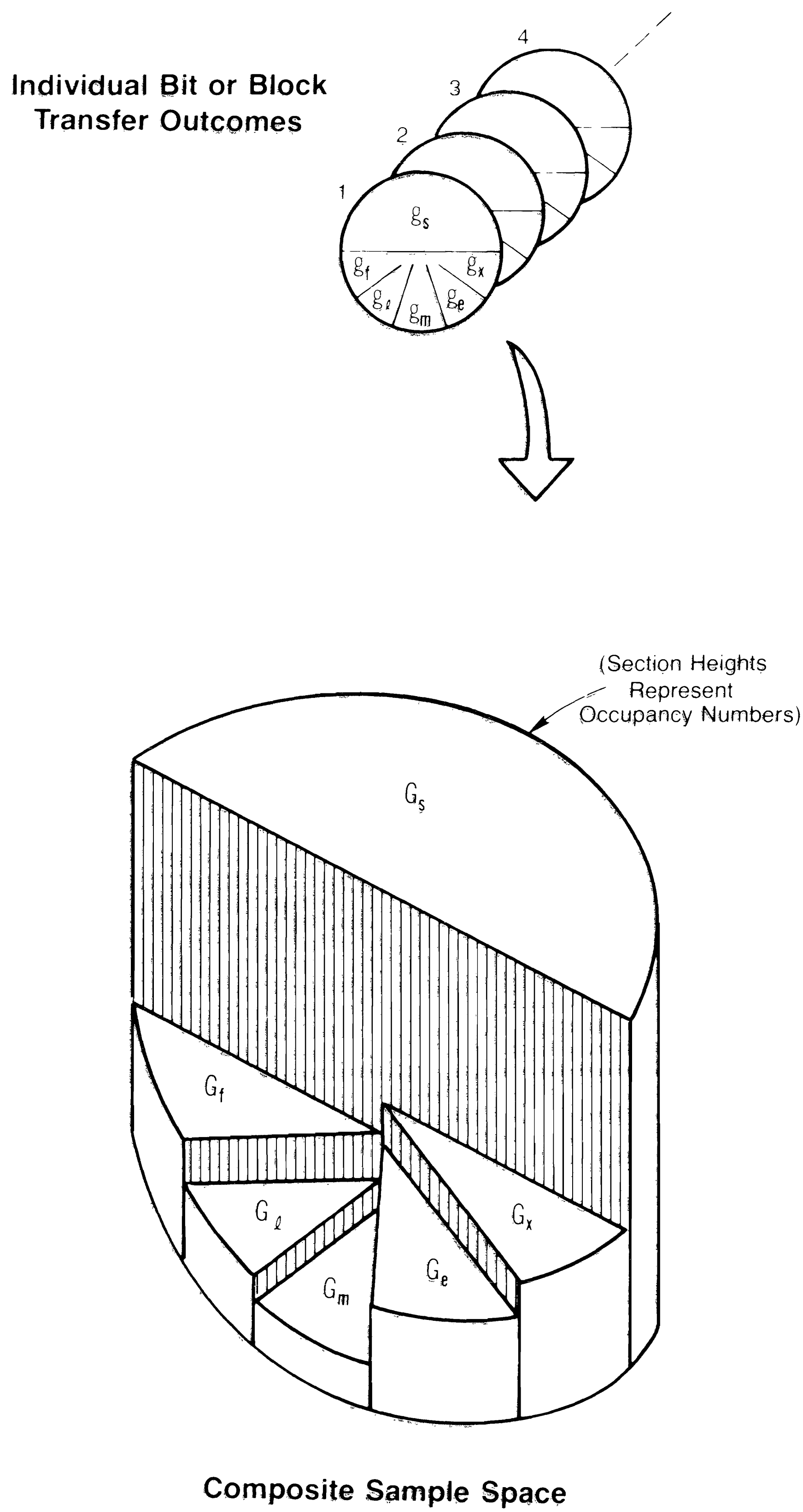


Figure 3.6. General sample space for a multi-event function.

We turn now to the second type of performance observation that can be made during a trial performance of a primary function g: observation of the time of occurrence of its component interface events. For simplicity, we consider only two event times in defining the primary outcomes: the time the function starts, t(g); and the time the function ends, t(g*). The difference between these discrete times is the "waiting time" to completion of the function, w(g*); i.e.,

$$w(g^*) = t(g^*) - t(g).$$

Given the assumptions of discrete time and a finite maximum performance period, we can represent the set of possible waiting times to completion of a primary function in a discrete, finite sample space W, as follows:

$$W = \{w_j | j=1,2,3,\ldots.m\}.$$

The variable j is a "waiting time index" which numbers the m discrete time intervals within the primary function performance period.

The primary performance parameters are defined on the basis of a composite sample space C which is the cross product of the event outcome and waiting time sample spaces defined above:

$$C = G \times W = \{g_i, w_j\}.$$

Each individual outcome in this composite sample space represents a particular combination of an event outcome (g_i) and a waiting time outcome (w_j).

The significance of this composite sample space can be summarized as follows. All possible outcomes of an individual performance of a defined primary function are represented by one and only one sample point (g_i, w_j) in the composite space. Therefore, any individual performance of the function can be completely described by specifying the single sample point which corresponds to its outcome; and a population of n performance trials can be completely described by specifying the corresponding n sample points. Each sample point can (in concept) be assigned a probability P(i,j), such that

$$0 \leq P(i,j) \leq 1$$

and

$$\sum_{i,j} P(i,j) = 1.$$

The set $\{P(i,j)\}$ then contains all significant information about performance of the specified function, and no other information; and represents a resolution of this information into "elementary units" which cannot be further subdivided. This organization of the "raw data" describing the performance of a function represents an ideal starting point for the development of performance parameters. A discussion of performance parameter classes which can be derived from this general sample space is presented in Section 3.4.

3.3.3 Selected Sample Spaces

We are now in a position to define the specific sample spaces selected to represent the five primary functions. Each sample space can be regarded as a special case of the composite sample space $\{g_i, w_j\}$; but since the waiting time sample spaces have the same form in each case, only the event outcomes are explicitly described. A distribution of possible waiting times $\{w_j\}$ associated with each event outcome is always implied.

The sample spaces are accompanied by outcome tables which identify the relevant event observations and define their specific influence on the function outcomes. Table 3.2 previews the primary sample spaces by identifying the event outcomes included in each. The g subscripts in this table are defined in Figure 3.5.

3.3.3.1. Access Sample Space

Figure 3.7 shows the outcome table and sample space for the access function. Five performance variables are included in the outcome table:

1. Access Responsibility Indicator (r_a). This variable identifies the entity responsible for access failures due to non-performance, as described in Section 5.

Table 3.2. Primary Function Outcomes

PRIMARY FUNCTIONS	OUTCOMES INCLUDED IN SAMPLE SPACE					
	g_s	g_e	g_m	g_ℓ	g_f	g_x
ACCESS	√		√	√	√	
BIT TRANSFER	√	√	√	√	√	√
BLOCK TRANSFER	√	√	√	√	√	√
MESSAGE TRANSFER	√	√	√	√	√	√
DISENGAGEMENT	√			√	√	

OUTCOME VARIABLES					OUTCOME
r_a	COMT	1B	SBS	UBS	
X	1*	1	X	X	a_s
X	0	1	X	X	a_m
X	X	0	1	0	a_ℓ
0	X	0	0	0	a_ℓ
X	X	0	0	1	a_f
1	X	0	0	0	a_f

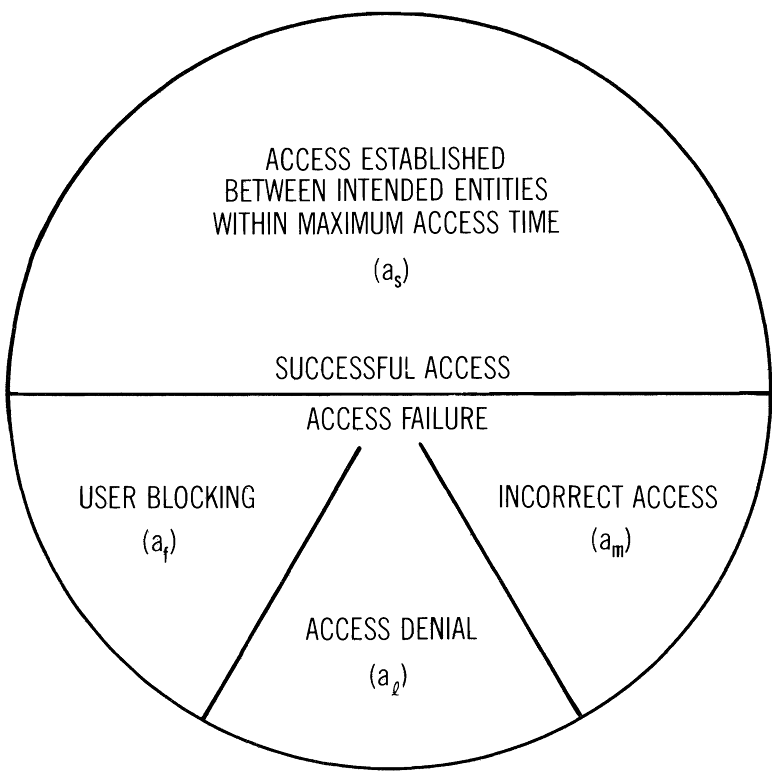
Legend

COMT, 1B, SBS, and UBS Variables:
1 = Event Occured
0 = Event Did Not Occur
X = Event Has No Influence on Outcome

r_2 Variable:
1 = User Responsible
0 = System Responsible
X = No Influence

*Not Required in the Case of Message-Oriented Systems

a. Outcome Table



b. Sample Space

Figure 3.7. Access outcome table and sample space.

2. Non-Originating User Commitment (COMT). This variable identifies whether the model event "non-originating user commitment" did or did not occur during the access performance period. COMT is relevant only in the case of circuit-oriented systems.
3. First Bit Transfer (1B). This variable identifies whether the start of user information transfer did or did not occur within the access performance period.
4. System Blocking Signal (SBS). This variable identifies whether the telecommunication system did or did not issue a blocking signal to one of the two users during the access performance period.
5. User Blocking Signal (UBS). This variable identifies whether a user did or did not issue a blocking signal to the telecommunication system during the access performance period.

Issuance of a blocking signal is evidenced by a Committed/Active (011) to Closing (10X) Transaction state transition on the part of the issuing entity.

These five performance variables define four event outcomes for the access function, as shown in the outcome table and descibed below:

1. Successful Access (a_s). Both COMT and 1B occur within the specified maximum access time. The variables r_a, SBS, and UBS have no influence on the access outcome in this case.
2. Incorrect Access (a_m). Event 1B occurs, but event COMT does not occur, within the maximum access time. Incorrect Access is identified by the negative event COMT=0 in order to avoid the necessity of examining all possible destination user locations for a "misdelivered" commitment request.

3. System Blocking (a_ℓ). This outcome can occur in two ways: (a) by occurrence of an SBS signal which supplants 1B; and (b) by the combined event 1B=0, r_a=0 in the case where no blocking signals are issued.

4. User Blocking (a_f). This outcome can occur in two ways: (a) by occurrence of a UBS which supplants 1B; and (b) by the combined event 1B=0, r_a=1 in the case where no blocking signals are issued.

In message-oriented systems, COMT occurs outside the access performance period and thus has no influence on any of the access outcomes. The implication of this is that Incorrect Access outcomes cannot occur in message-oriented systems. The possible outcomes of message-oriented access can still be represented by the sample space of Figure 3.7, by regarding a_m as a "null outcome," i.e., $P(a_m)=0$.

It is possible for users of a message switched system to be incorrectly connected to an unintended system entity during the access process. An example is the case where the user must connect to the first store-and-forward switching center via the public (circuit) switched network. Although such "incorrect connection" events could be distinguished from access denial in some cases, it was decided that they should not be defined as separate outcomes since their detection would be system dependent. The incorrect content (g_e) and extra event (g_x) outcomes are not relevant in the case of the access function.

3.3.3.2. User Information Transfer Outcomes

Section 2.5 pointed out the need for a "standard data correlator" capable of comparing source user output and destination user input records to associate corresponding transmitted and received bits. We assume the feasibility of such an algorithm as the basis for defining the user information transfer performance outcomes. Figure 3.8 illustrates the general nature of the process we envision. The correlator accepts as inputs a source user output record and a destination user input record to be compared. The SU output record is depicted as a sequence of

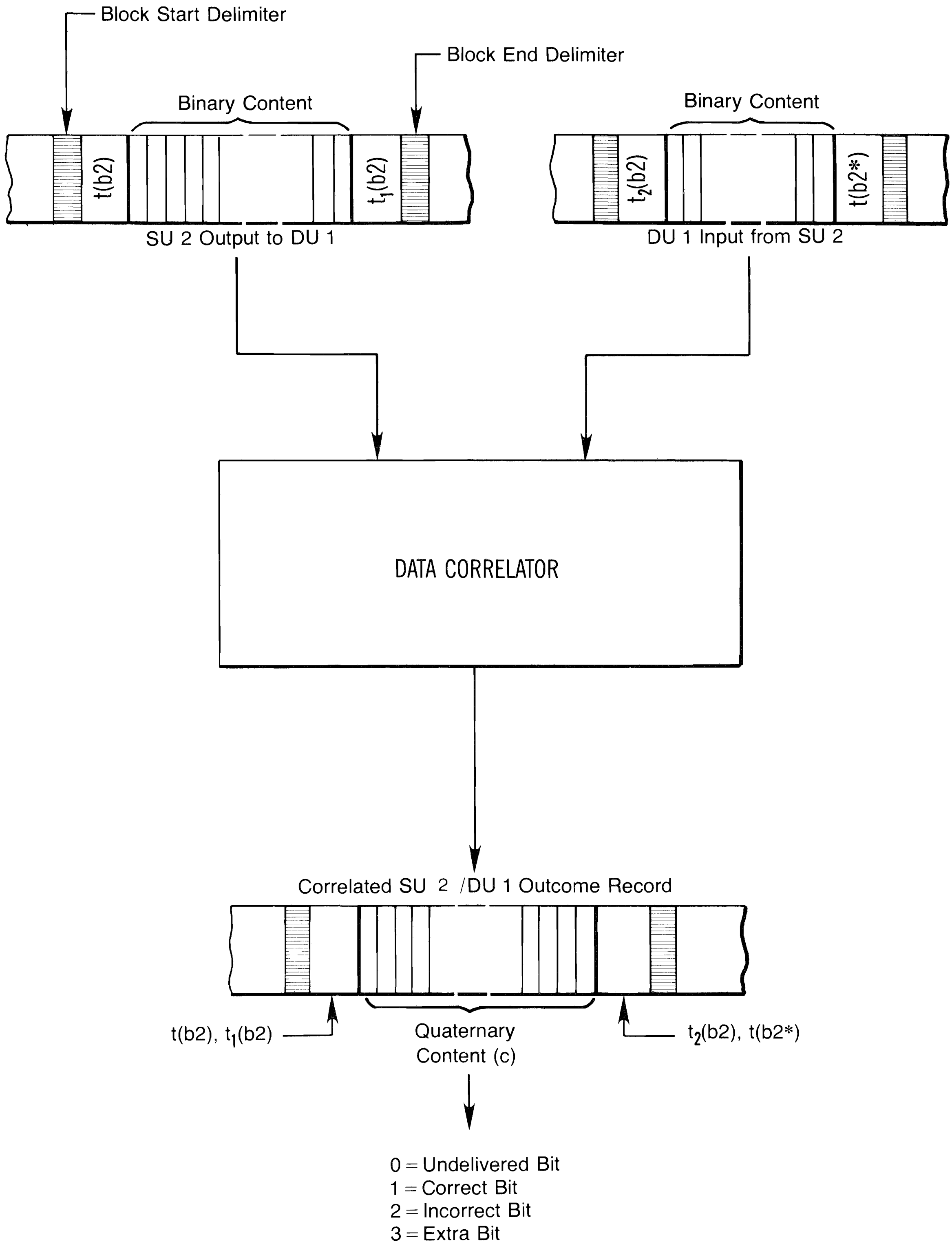
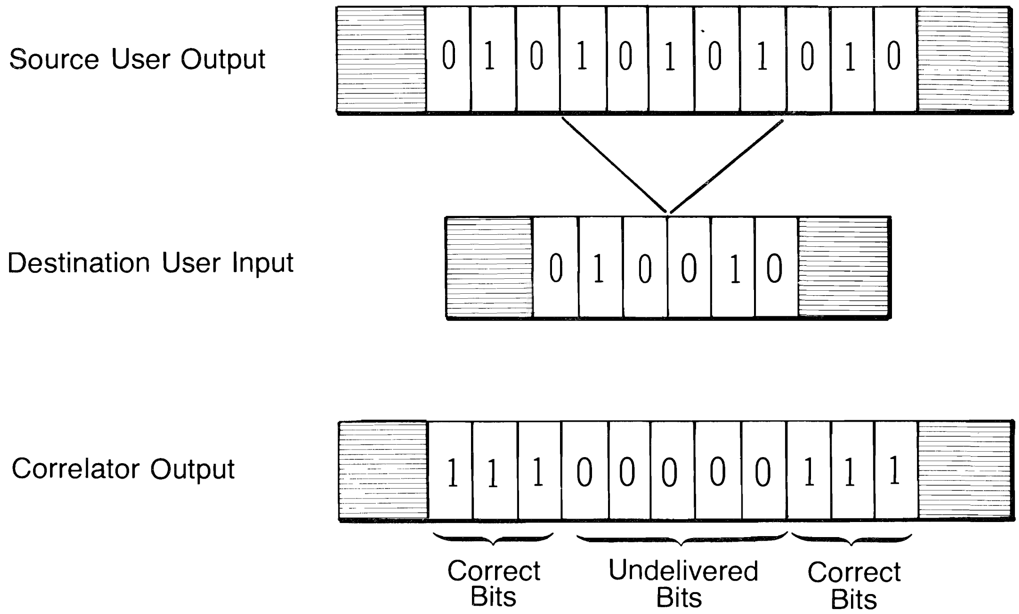


Figure 3.8. Data correlator input/output formats.

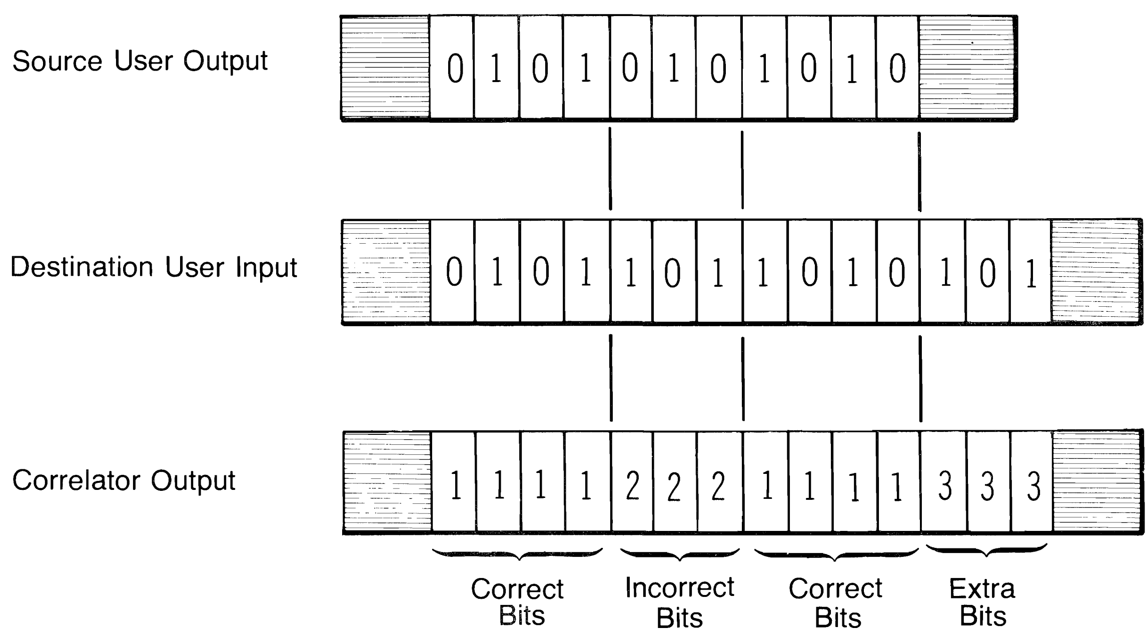
delimited blocks, each identified by an output start time t(b2) and an output end time t_1(b2). The DU input record is identical in form, except that the start and end times, t_2(b2) and t(b2*), identify the start and end of input rather than output. Both the source user output record and the destination user input record are associated with a particular source/destination pair, as illustrated in Figure 2.8.

In order to provide a basis for defining the user information transfer performance outcomes, the correlator would compare each transmitted block, bit by bit, with the blocks received at the intended destination between t(b2) and t(b2) plus the maximum block transfer time. The correlator would produce as output a sequence of correlated blocks, each comprising a set of quaternary symbols (c=0,1,2,3) identifying the outcome of the matching process for each bit in the input and output records. Output bits not detectable in the input record would be identified as Undelivered Bits (c=0); output bits present in the input record (and correct in content) would be identified as Correct Bits (c=1); output bits present in the input record (but incorrect in content) would be identified as Incorrect Bits (c=2); and input bits not detectable in the output record would be identified as Extra Bits (c=3). The correlator would also associate the four input/output event times with each correlated block. In the case where all bits in an n-bit source block were transferred to the intended destination without error, the correlated block would contain n 1's. Figure 3.9 illustrates the data correlator output under various error conditions.

It is envisioned that the primary use of the data correlator will be in comparing a source user output record (e.g., 0_{12} in Fig. 2.8) with the corresponding intended destination user input record (e.g., I_{12}). Nevertheless, a second type of comparison is required if "misdelivered" bits (and blocks) are to be distinguished from extra ones: comparison of all Extra Bits identified in the first (source/intended destination) comparison with the source user outputs to other destination users (e.g.,



a. Example 1: Undelivered Bits within a Block



b. Example 2: Incorrect and Extra Bits within a Block

Figure 3.9. Data correlator operation examples.

the output records $O_{12}, O_{13}, O_{14} \ldots O_{1m}$ in Fig. 2.8). Any Correct Bit (1) or verifiable Incorrect Bit (2) outputs from this second comparison would be redefined as misdelivered bits relative to the first user pair (Source 1/Destination 2). This is logical since correlator outputs (1) and (2) indicate successful matching of output and input bits; when this occurs in comparing a source user output record with the input record of an unintended destination, it indicates misdelivery.

It is recognized that the latter type of comparison would be difficult and time-consuming in a large multi-user network. It is described here primarily to provide a basis for defining the misdelivery outcomes. If such comparisons are not undertaken in a measurement situation, misdelivery outcomes will simply be counted as "extra" bits or blocks. This method of defining misdelivery outcomes has two advantages:

1. Measurability. Transmissions are checked for possible misdelivery only when Extra Bits are received. If misdelivery were defined on the basis of transmitted data, it would be necessary to confirm the absence of every transmitted block at every potential unintended destination in order to ensure that misdelivery had not occurred. If misdelivery were defined on the basis of undelivered data, instances in which a message was delivered both to the intended destination and to another (unintended) destination would not be properly detected.
2. Specificity. All misdelivery outcomes are associated with a specific source/destination pair: the originator and (unintended) recipient of the misdelivered information. Other, more general types of misdelivery outcomes (such as misdelivery of any bit output by a source, independent of destination) can be derived by combining appropriate source/destination outcomes.

The suggested approach also facilitates translation of user-to-user performance values into corresponding values for a multi-user network.

We are now in a position to define the possible outcomes for the bit transfer function (Fig. 3.10). Two outcome variables are employed (in addition to the correlator output, c, associated with each transferred bit):

1. Comparison Variable (CPR). This variable specifies the relationship between the source and destination records being compared. The variable CPR is set to 1 to identify the correlator output being examined as the result of a "normal" comparison, between corresponding source user output and (intended) destination user input records. The variable CPR is set to 0 to identify the second type of comparison described above; i.e., comparison of Extra Bits identified by a source/ intended destination comparison with source user outputs to other destination users.

2. Block Transfer Responsibility Indicator (r_{b2}). This variable is used to attribute responsibility for undelivered bits to the system or the users, as described in Section 5. Note that responsibility for undelivered bits is determined by reference to the responsibility indicator for the associated block.

These variables define six event outcomes for the bit transfer function:

1. Successful Bit Transfer (bl_s). This outcome is identified by the joint occurrence CPR=1, c=1.
2. Incorrect Bit (bl_e). This outcome is identified by the joint occurrence CPR=1, c=2.
3. Misdelivered Bit (bl_m). This outcome is identified by the joint occurrence CPR=0, c=1 or 2.
4. Lost Bit (bl_ℓ). This outcome is identified by the joint occurrence CPR=1, c=0, r_{b2}=0.
5. Refused Bit (bl_f). This outcome is identified by the occurrence CPR=1, c=0, r_{b2}=1.
6. Extra Bit (bl_x). This outcome is identified by the occurrence c=3 (for both CPR=0 and CPR=1).

OUTCOME VARIABLES			OUTCOME
CPR	r_{b2}	c	
1	X	1	bl_s
1	X	2	bl_e
0	X	1 or 2	bl_m
1	0	0	bl_ℓ
1	1	0	bl_f
0, 1	X	3	bl_x

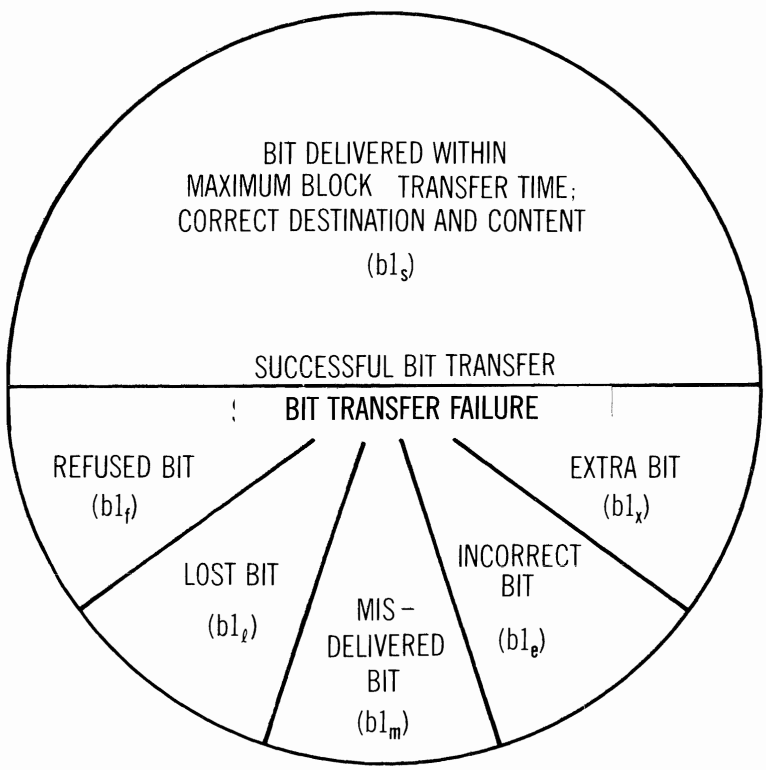
Legend

CPR Variable:
- 1 = Normal Comparison
- 0 = Extra Bit Comparison

r_{b2} Variable:
- 1 = User Responsible
- 0 = System Responsible
- X = No Influence

c Variable:
- 0 = Undelivered Bit
- 1 = Correct Bit
- 1 = Incorrect Bit
- 3 = Extra Bit

a. Outcome Table



b. Sample Space

Figure 3.10. Bit transfer outcome table and sample space.

All six event outcomes identified in Figure 3.5 are thus relevant in the case of the bit transfer function.

Figure 3.11 shows the outcome table and sample space for the block transfer function. The outcome table contains a separate outcome variable, b_i, for each bit transfer outcome included in the correlated block. The block transfer outcomes are determined solely on the basis of the component bit transfer outcomes. The block outcome vectors are reduced to the same six outcomes defined in the bit transfer case, as indicated in the outcome table and described below:

1. Successful Block Transfer ($b2_s$). This outcome is identified by joint occurrence of outcome $b1_s$ for all bit transfer outcomes in the delimited block.
2. Incorrect Block ($b2_e$). This outcome includes all possible combinations of bit transfer outcomes not included in (1) above and (3-6) below.
3. Misdelivered Block ($b2_m$). This outcome is identified by the occurrence $b1_m$ for any bit transfer outcome in the delimited block.
4. Lost Block ($b2_\ell$). This outcome is identified by the joint occurrence $b1_\ell$ for all bit transfer outcomes in the delimited block.
5. Refused Block ($b2_f$). This outcome includes all occurrences in which one or more bit transfer outcomes in the delimited block are $b1_f$.
6. Extra Block ($b2_x$). This outcome is identified by the joint occurrence $b1_x$ for all bit transfer outcomes in the delimited block.

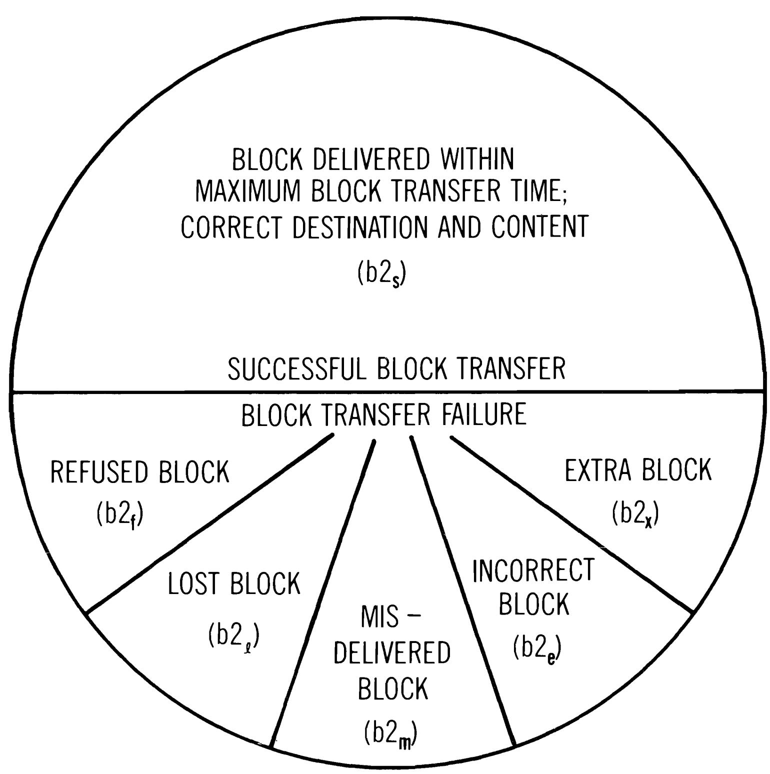
Two separate sample spaces are defined to describe the message transfer outcomes: a bit-oriented sample space (Fig. 3.12a) and a block-oriented sample space (Fig. 3.12b). In each sample space, the individual outcomes comprise a sequence of six occupancy numbers, each number representing the total number of bit (or block) transfer attempts encountering each bit (or block) transfer outcome (Sec. 3.3.2). Individual bit and block transfer

OUTCOME VARIABLES						OUTCOME
b_1	b_2	• • •	b_i	• • •	b_n	
1	1	• • •	1	• • •	1	$b2_s$
(ALL b_i COMBINATIONS NOT INCLUDED UNDER OTHER OUTCOMES)						$b2_e$
b_i = 3 (FOR ONE OR MORE DIGITS, i)						$b2_m$
4	4	• • •	4	• • •	4	$b2_\ell$
b_i = 5 (FOR ONE OR MORE DIGITS, i)						$b2_f$
6	6	• • •	6	• • •	6	$b2_x$

Legend

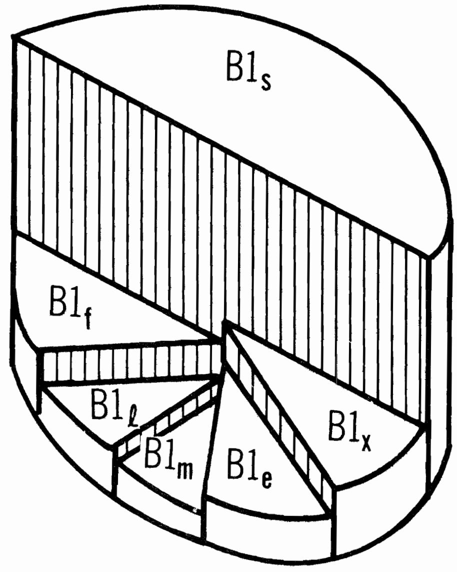
b_i Value	Bit Transfer Outcome
1	$b1_s$ —Successful
2	$b1_e$ —Incorrect
3	$b1_m$ —Misdelivered
4	$b1_\ell$ —Lost
5	$b1_f$ —Refused
6	$b1_x$ —Extra

a. Outcome Table

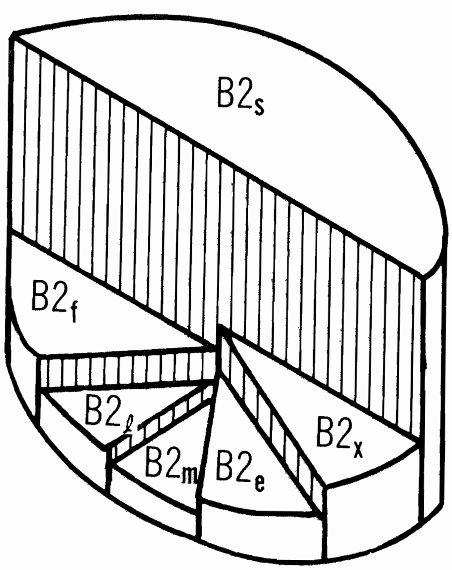


b. Sample Space

Figure 3.11. Block transfer outcome table and sample space.



a. Bit-Oriented Message Transfer Sample Space



b. Block-Oriented Message Transfer Sample Space

Figure 3.12. Message transfer sample spaces.

outcomes are determined as specified in the preceding paragraphs. No outcome table is shown for the message transfer function, since the occupancy number sequences are not reduced to simpler outcomes in defining the primary message transfer parameters.

The message transfer sample spaces are used in defining certain long-term user information transfer parameters (e.g., transfer rates); and represent the sample size to be used in calculating relative-frequency estimates for the bit transfer and block transfer parameters. The bit-oriented message transfer sample space also provides a basis for defining the secondary parameters (Sec. 4).

3.3.3.3. Disengagement Sample Space

Figure 3.13 shows the outcome table and sample space for the disengagement function. Two performance variables are included in the outcome table:

1. Disengagement Responsibility Indicator (r_d). This variable identifies the entity responsible for disengagement failures, as described in Section 5.
2. Disengagement Confirmation Variable (DC). This variable indicates whether the model event "disengagement confirmation" did or did not occur within the disengagement performance period.

These variables define three event outcomes for the disengagement function:

1. Successful Disengagement (d_s). Event DC=1 occurs within the specified maximum disengagement time. The variable r_d has no influence on the disengagement outcome in this case.
2. Disengagement Denial (d_ℓ). This outcome is defined by the joint occurrence DC=0, r_d=0; i.e., disengagement failure for which the system is responsible.
3. User Disengagement Blocking (d_f). This outcome is defined by the joint occurrence DC=0, r_d=1; i.e., disengagement failure for which the user is responsible.

This completes the definition of sample spaces for the primary functions.

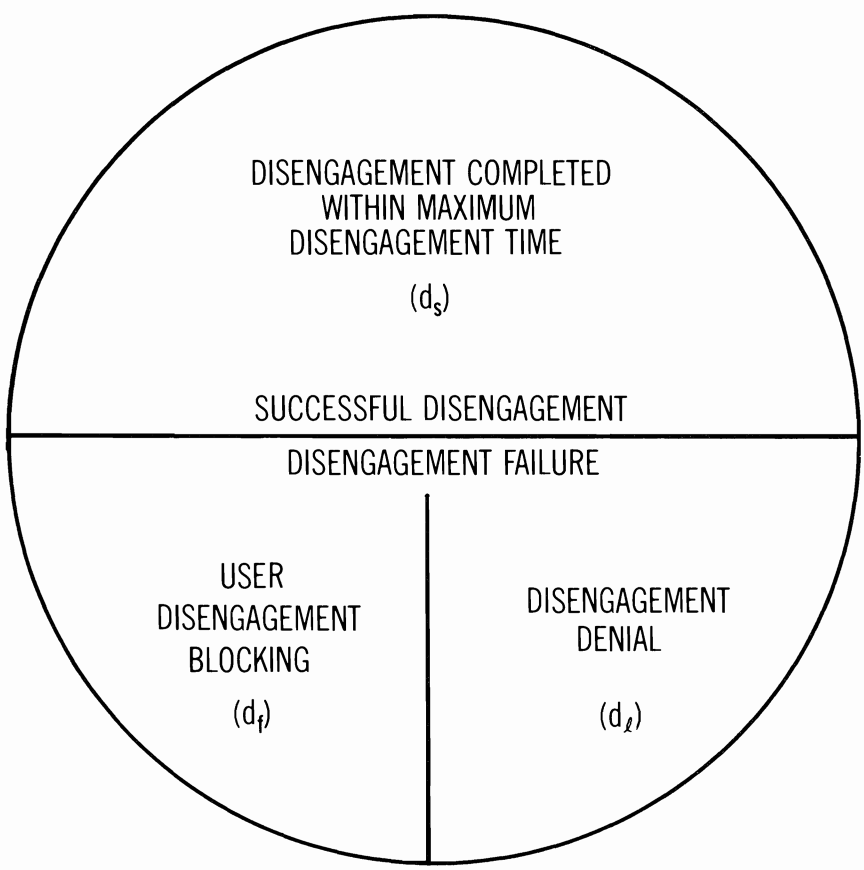
OUTCOME VARIABLES		OUTCOME
r_d	DC	
X	1	d_s
0	0	d_ℓ
1	0	d_f

Legend

DC Variable:
- 1 = Event Occurs (Or Indicator Is Positive)
- 0 = Event Does Not Occur (Or Indicator is Negative)
- X = Event Has No Influence on Outcome

r_d Variable:
- 1 = User Responsible
- 0 = System Responsible
- X = No Influence

a. Outcome Table



b. Sample Space

Figure 3.13. Disengagement outcome table and sample space.

3.4. Parameter Selection

The preceeding subsection defined a set of sample spaces, of the general form $\{g_i, w_j\}$, to represent the five primary functions; and pointed out that all relevant information describing the performance of a given function can be represented in an associated set of elementary outcome probabilities $\{P(i,j)\}$. It follows that any parameter for describing performance relative to a given function can be defined as a mathematical rule or procedure for summarizing this set of probabilities. Our purpose in this subsection is to survey the various ways of accomplishing this summarization; and to define the primary performance parameters actually selected for inclusion in the proposed Federal Standard.

3.4.1. Parameter Options

In considering the possible data summarization approaches, we begin with the simplest case, in which the experiment is characterized as a "single trial" having only one event outcome. This experiment is represented by the event sample space shown in Figure 3.5, with the associated distribution of possible waiting times $\{w_j\}$ implied. There are two general approaches which can be used to define performance parameters on this sample space:

1. Partition the sample space into disjoint subsets of sample points; add the corresponding sample point probabilities, to produce "aggregate" outcome probabilities $P(I,J)$ for the subsets; and define these subset probabilities directly as performance parameters.
2. Define random variables (RV's) on the sample space; and compare the corresponding outcome probabilities as ordered sets or probability distributions. The various moments of probability distributions can then be defined as performance parameters.

The first approach is the most appropriate for summarizing the event outcomes g_i, since the numbering of the possible events, i, is arbitrary. The second approach is the most appropriate for summarizing the performance time outcomes w_j, since the possible performance times naturally form an ordered set in j.

As a simple example of the first parameter definition approach, the sample space $\{g_i, w_j\}$ could be divided into two subsets: one containing all sample points in which the intended event (say g_1) occurs, independent of performance time; and the other containing all remaining points. This partition would define two very simple probability parameters, the probability of successful performance $P(g_1)$ and its complement, $1-P(g_1)$.

As an example of the second parameter definition approach, we could assign each sample point (i,j) a numerical value equal to j, the (discrete) performance time (Sec. 3.3.2). This defines a random variable J which represents the "response time" of the function, irrespective of the response event, i. Note that in all but the special case where every sample point (i,j) is assigned a different RV value, the definition of an RV also partitions the sample space, by grouping all sample points having the same RV value. The probability associated with an RV value is thus, in general, also a sum of individual sample point probabilities. The probability distribution of a random variable J is represented by the symbol f(j).

The two data summarization approaches described above can also be combined, serially, to define <u>conditional</u> performance parameters. For example, we could partition the sample space $\{g_i, w_j\}$ into two subsets as indicated in (1) above, and then take the subset of points $\{g_i w_j | i \neq 1\}$ as a new sample space (subspace). Performance parameters could then be defined for this sub-space, either by partitioning or by defining random variables. Parameters defined in this way are termed "conditional" because they depend on, and are only defined for, a specified subset of sample points.

These alternatives provide a basis for defining five general classes of parameters which can be used to describe performance relative to a single-trial function:

1. Event Probabilities P(i). These parameters are formed by summing elementary outcome probabilities over all output times j; i.e.,

$$P(i) = \sum_j P(i,j)$$

These probabilities express the likelihood that a particular event outcome i will occur on a given trial, irrespective of performance time. The probability of successful performance is one parameter in this class, as was noted above.

2. Performance Time Variables Y(j). The performance time probability distribution f(j) expresses the likelihood that a trial will end at a particular performance time j, irrespective of the event outcome i. Each probability in the distribution is formed by summing outcome probabilities over all i; i.e., $f(j)=\sum_i P(i,j)$.

Although the distribution f(j) can be summarized in many ways, the most commonly used statistics are the average or expected value

$$E(j) = \sum_{j=1}^{m} jf(j)=\mu_j,$$

the variance

$$Var(j) = E[(j-\mu_j)^2],$$

and the standard deviation

$$\sigma_j = \sqrt{Var(j)}.$$

Each of these statistics represents a possible performance parameter.

3. Conditional Event Probabilities P(i|I). These parameters are formed by a double application of the summation indicated in (1) above; i.e.,

$$P(i|I) = \frac{P(i)}{P(I)} .$$

I denotes a specified group of event outcomes; i denotes a particular outcome within that group.

$P(i|I)$ expresses the likelihood of i given I. A simple example of a parameter in this class is the probability of incorrect performance given performance failure.

4. Conditional Performance Time Variables $Y(j|i)$. These parameters are functions of the random variable j, defined on a subset of the sample space corresponding to a particular event outcome, i. Each probability in the distribution $f(j|i)$ is a conditional probability of the form

$$P(j|i) = \frac{P(i,j)}{P(i)} .$$

Again, the distribution $f(j|i)$ can be summarized by various statistics. One which is particularly useful is the conditional mean, $E(j|i)$; this can be described more informally as the "waiting time" to a particular outcome, $W(i)$.

5. Conditional Event Variables $Y(i|j)$. A third type of conditional probability that can be defined on the sample space $\{g_i, w_j\}$ is the conditional probability of i given j, i.e.:

$$P(i|j) = \frac{P(i,j)}{P(j)} .$$

This conditional probability can be regarded as a random variable defined on the sub-space described in (4) above, since it is a function which assigns a value to every sample point in that space. The distribution of $P(i|j)$, for a fixed i, is given by

$$P_i(j) = \frac{P(i,j)/P(j)}{\sum_j P(i,j)/P(j)}$$

Statistics describing this distribution would give information about how the likelihood of a particular event outcome varies with performance time.

A simple analogy will clarify the significance of this latter type of probability (Nesenbergs, M., private communication).

Assume that the weather at a particular geographical location is summarized in terms of one of four discrete states g_i (g_1=rain, g_2=cloudy, g_3=snow, g_4=clear), on each day of the year j. The set of probabilities $P_1(j)$ then represents a distribution of the probability of rain at that location as a function of the time of year. The numerator of the fraction is the probability of rain on any given day of the year, and the denominator is the total of all the individual rain probabilities, summed over every day of the year. A comparison of (e.g.) the mean value of this distribution for different locations would provide some useful information about the variation in climate between the two locations.

We now consider the case in which the function to be described comprises a succession of "repeated trials" of a single-event function; i.e., the transfer of a user information "message" comprising many individual bits or blocks. This experiment is represented by the general sample space shown in Figure 3.6, with a distribution of possible waiting times $\{w_j\}$ implied. Again, each sample point g_i in the multi-event sample space comprises a set of six "occupancy numbers," $\{G_1G_2G_3G_4G_5G_6\}$, each representing the total number of bit (or block) transfer trials encountering a particular outcome. As in the single-event case, there are two general approaches to the definition of performance parameters: (1) partition the sample space, and define the composite outcome probabilities P(I,J) as performance parameters; and (2) define random variables on the sample space, and select particular moments of the associated probability distributions as performance parameters.

Each of the five single-trial parameter classes defined above can also be applied to the repeated-trial functions; but in the latter case, there are additional possibilities as well: the total number (G_i) of single-event outcomes of a particular type (i) can be used to define random variables. The four most useful ways of defining RV's based on G_i are summarized below.

1. The number G_i itself can be defined as a random variable. If the total number of trials in the experiment

(e.g., bits in a message) is n, this approach partitions the sample space into n subsets, one associated with each possible G_i value. The probability distribution $f(G_i)$ can then be summarized in terms of moments. One parameter in this class which is very commonly used in error control work is P(m,n), the probability of m errors in a block of n bits. This parameter is based on an assumption of only two possible bit transfer outcomes: bit error (1) or no bit error (0). Specific probabilities in the distribution P(m,n) can also be defined as performance parameters; for example, [1-P(0,n)] is the block error probability, and [1-P(0,1)] is the bit error probability.

2. The number G_i can be expressed as a proportion of the total number of single-event trials (or an appropriate subset of this total); e.g.,

$$\hat{P}(i) = \frac{G_i}{\sum_i G_i} .$$

Proportions of this form are very significant from a measurement point of view, since they provide unbiased <u>relative frequency estimates</u> of the corresponding probabilities P(i). All of the probability parameters defined in the proposed Federal Standard are defined and measured in terms of proportions of this type. The relationship between the number of trials, the number of observed outcomes, and the accuracy of $G_i/\sum G_i$ as an estimate of P(i) is described in Crow (1974) and Crow and Miles (1977).

3. The number G_i can be divided by the total performance time, w_j, to produce a <u>time rate</u>:

$$R(i) = \left[\frac{G_i}{w_j}\right]$$

Bit transfer rate (also known as "throughput") is a commonly used parameter in this class. Every sample point in the (partitioned) sample space $\{G_i, w_j\}$ defines a particular value for R(i); any measured value of the ratio (G_i/w_j) is thus a relative frequency estimate of the "true" value of R(i).

4. The number G_i can be divided by the maximum number of <u>possible</u> occurrences of G_i during the performance period, i.e., G_i(max). Note that this number differs in general from the number of actual or observed single-event trials, $\sum_i G_i$, since event outcomes may not occur at the maximum possible rate. A commonly used parameter in this class is "rate efficiency," or loosely, "efficiency."

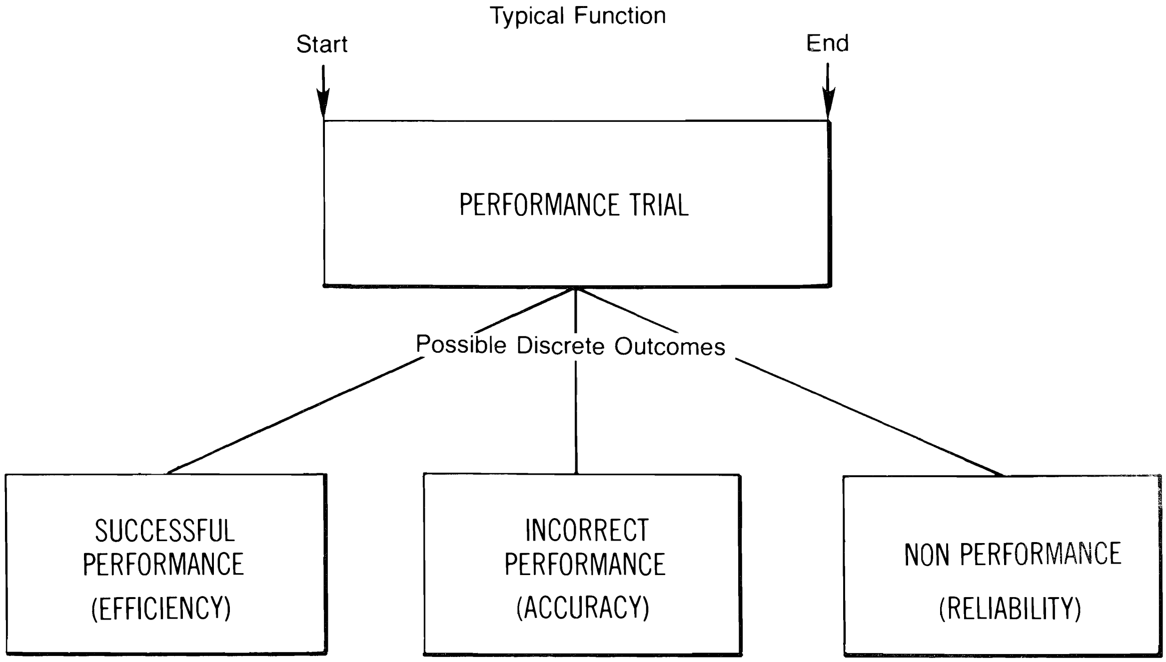
The five single-trial and four repeated-trial parameter classes defined above provided the basic list of options from which performance parameters were selected for the proposed Federal Standard.

3.4.2. Selected Parameters

Of the four desired parameter attributes listed in Section 1.1, three had a direct influence on the parameter selection process: user orientation, simplicity, and completeness. The first of these required that parameters be selected on the basis of expressed user needs. Comments of Federal Government user and supplier agencies were given particular consideration, since these organizations will have primary responsibility for implementing the proposed standard.

The objective of simplicity required that selected parameters be as fundamental as possible, so that (1) their significance would be readily apparent to users, and (2) their values could be determined with a minimum of user effort. The parameters included in the proposed Federal Standard were chosen from among four of the above classes: conditional probabilities, waiting times, time rates, and rate efficiencies.

The objective of completeness was addressed in the manner illustrated in Figure 3.14. In order to ensure completeness in



a. Performance Outcome Categories

FUNCTION	PERFORMANCE CRITERION		
	EFFICIENCY	ACCURACY	RELIABILITY
ACCESS	• PROBABILITY • TIME	• PROBABILITY • TIME	• PROBABILITY • TIME
BIT TRANSFER	• PROBABILITY • TIME	• PROBABILITY • TIME	• PROBABILITY • TIME
BLOCK TRANSFER	• PROBABILITY • TIME	• PROBABILITY • TIME	• PROBABILITY • TIME
MESSAGE TRANSFER	• PROBABILITY • TIME	• PROBABILITY • TIME	• PROBABILITY • TIME
DISENGAGEMENT	• PROBABILITY • TIME	• PROBABILITY • TIME	• PROBABILITY • TIME

b. Performance Assessment Matrix

Figure 3.14. Parameter selection approach.

the selected parameter set, each of the five primary functions was considered from the point of view of each of the three defined performance criteria, in matrix fashion; and appropriate probability and time parameters were chosen wherever specific user concern with a particular function/criterion pair could be demonstrated.

The following paragraphs define the selected parameters in two ways:

1. By applying the general parameter definitions developed above to particular sample spaces and outcomes. For example, the general waiting time parameter W(i) is applied to the Successful Access (a_s) outcome to produce the performance parameter Access Time, $W(a_s)$.
2. By reference to parameter definition flowcharts and associated equations. These provide a procedure for calculating values for the defined parameters based on a measured sample population of performance trials.

The notation used in the sample space diagrams and flowcharts is a simple extension of that used in the preceding subsection. Each function is represented by a lower case mnemonic index (e.g., "a" for access). The various performance outcomes are represented by subscripted function indices (e.g., "a_s" for Successful Access). These lower case indices are used as counters in the parameter definition flowcharts; and the corresponding total numbers of trials and outcomes observed during a performance measurement are represented by upper case letters (e.g., "A" and "A_s"). Probabilities, average waiting times, time rates, and rate efficiencies are represented by the symbols P( ), W( ), R( ), and Q( ), respectively; individual event times are represented by the symbol t( ). The argument ( ) in each case is the event outcome of interest; for example, the expression $W(a_s)$ denotes the average waiting time to Successful Access. The lower case symbol w( ) is used to distinguish an individual waiting time value from the corresponding average W( ). Specified (or "nominal") parameter values are distinguished from the corresponding measured values by the subscript N; for example, the symbol $W_N(a_s)$ denotes the specified value for the parameter Access Time.

It was concluded that all system performance parameter values should be based on a reduced sample population excluding performance failures attributable to user non-performance. A prime mark (') after a function index or population total indicates that the symbol refers to the reduced population (and therefore excludes user dependent failures). For example, the index a' counts Successful Access, Incorrect Access, and Access Denial outcomes, but does not count User Blocking (a_f) outcomes; and the constant A' represents the total number of access attempts in a sample population, excluding the A_f which are attributable to User Blocking.

3.4.2.1. Access Parameters

In considering the access function, it was concluded that two of the five single-trial parameter classes defined in Section 3.4.1 were of primary significance to end users: the conditional event probabilities $P(i|I)$, for the subset (a') of trials excluding User Blocking; and the waiting times to specified outcome W(i), again excluding the outcomes a_f. The simple event probabilities P(i) and the performance time variables Y(j) were eliminated by the decision to exclude User Blocking outcomes from the sample population used in calculating system performance parameter values. The conditional event variables $Y(i|j)$ were excluded on the basis of their relative complexity and the lack of expressed user interest in them.

These decisions reduced the set of possible access parameters to six: one conditional probability and one waiting time associated with each of the three outcomes a_s, a_ℓ, and a_m. Three of these parameters were ultimately selected for inclusion in the proposed Federal Standard (Fig. 3.15): one parameter associated with each of the three general performance criteria. The waiting times associated with the failure outcomes were excluded on the basis of the relatively infrequent occurrence of these outcomes in most systems. The probability $P(a_s)$ was

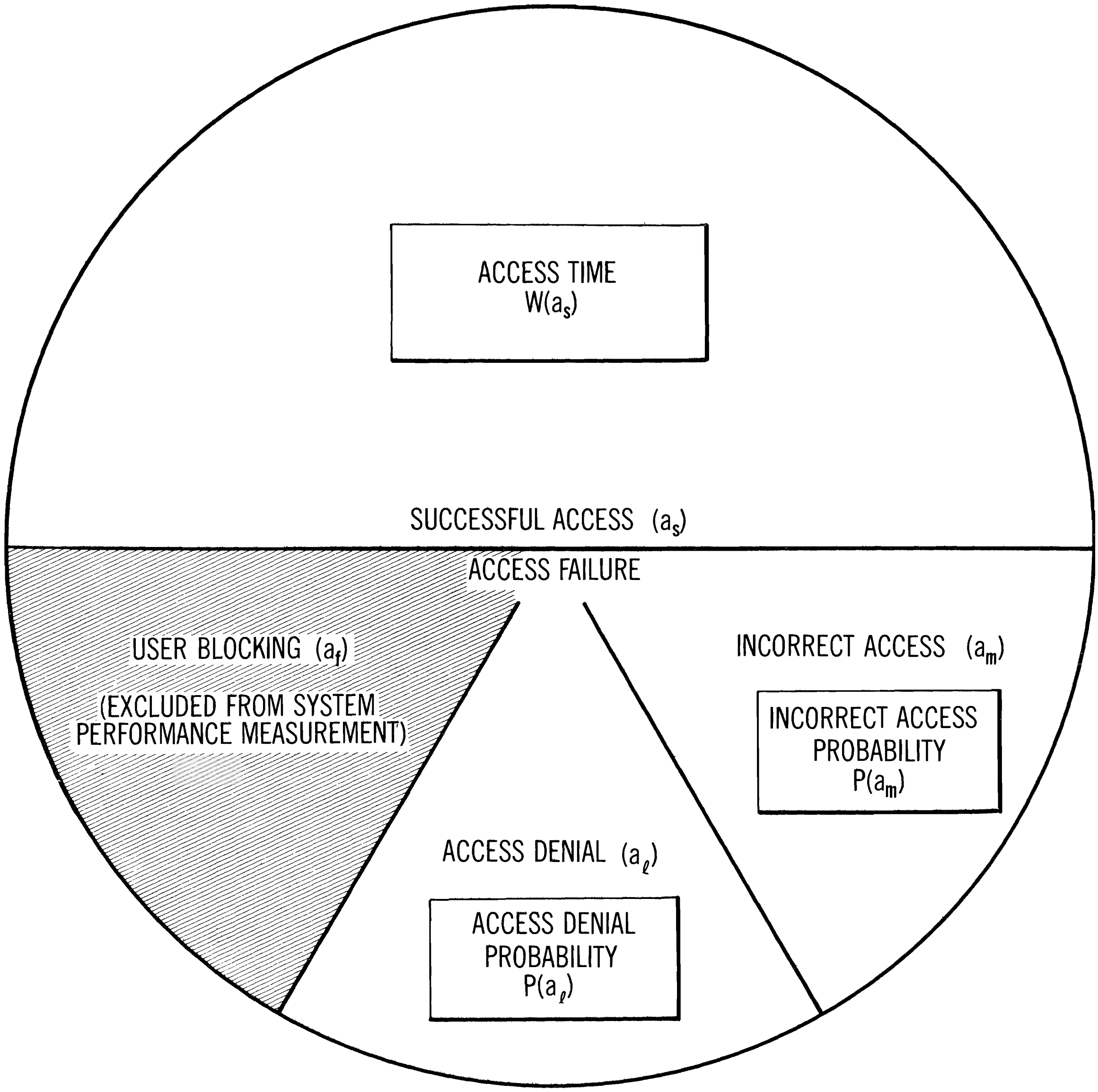


Figure 3.15. Selected access parameters.

excluded since it can be calculated from $P(a_\ell)$ and $P(a_m)$, and is therefore technically redundant with them. Narrative/symbolic definitions for the selected parameters are provided below.

<u>Access Time</u> $W(a_s)$ - Average value of elapsed time between the start of an access attempt $t(a)$ and Successful Access $t(a_s)$. Elapsed time values are calculated only on access attempts that result in Successful Access.

<u>Incorrect Access Probability</u> $P(a_m)$ - Ratio of total access attempts that result in Incorrect Access (A_m) to total access attempts included in the reduced sample population (A').

<u>Access Denial Probability</u> $P(A_\ell)$ - Ratio of total access attempts that result in Access Denial (A_ℓ) to total access attempts included in the reduced sample population (A').

The preceeding subsections have referred to a constant "maximum access time" which is used in identifying non-performance outcomes. This constant can now be defined more precisely: it equals three times the specified (nominal) value for the parameter Access Time; i.e., $3W_N(a_s)$. The maximum performance times for all primary functions are specified in this same way, as a function of the corresponding nominal waiting time values, in order to permit comparison of failure probabilities between systems having different specified performance times. Such comparisons would be less meaningful if a constant maximum performance time were selected. The choice of the constant 3 as a performance time multiplier was based on an analysis of various available communication performance time distributions (e.g., AT&T, 1971; Cole, 1971; Kleinrock, 1976; Duffy and Mercer, 1978). The objective was to select a minimum integer multiplier such that virtually all observed performance times would be below the specified maximum. The alternative of defining the maximum performance time as a function of the standard deviation of the various performance time distributions was discarded since this approach would favor systems with more variable performance.

Figure 3.16 is a parameter definition flowchart for the access function. The flowchart complements the access outcome table of Figure 3.7 by defining a logical sequence in which the

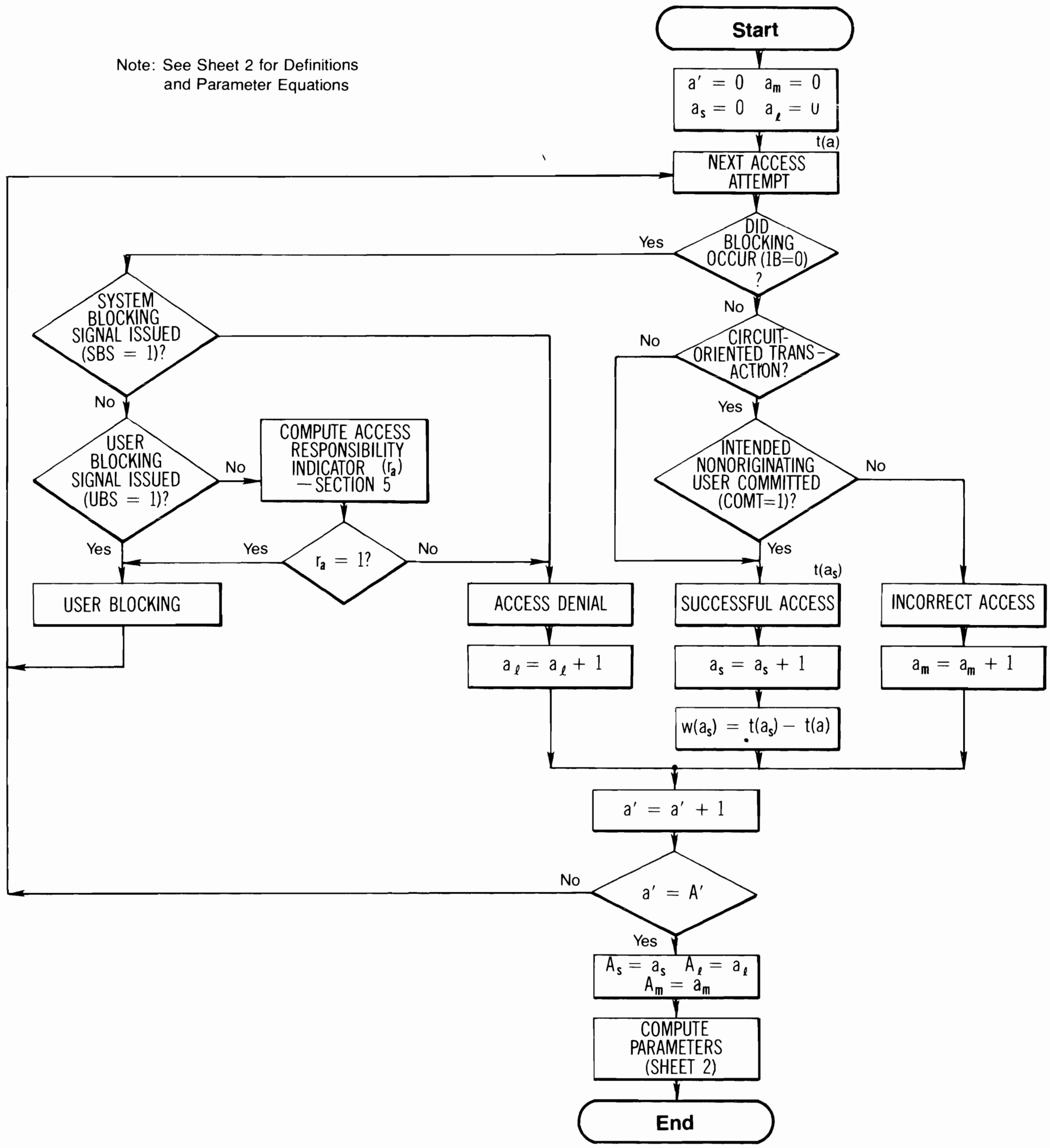


Figure 3.16 (sheet 1). Access parameter definition flowchart.

ACCESS PARAMETERS

1. Access Time $= W(a_s) = \frac{1}{A_s}\sum_{a_s=1}^{A_s} w(a_s)$

2. Incorrect Access Probability $= P(a_m) = A_m/A'$

3. Access Denial Probability $= P(a_\ell) = A_\ell/A'$

DEFINITION OF CONSTANTS

A' = Total number of access attempts counted during an access parameter measurement.

DEFINITION OF VARIABLES

a' = Access attempt counter. The count excludes attempts that fail due to user blocking.

a_s = Successful Access outcome counter.

A_s = Total number of successful Access outcomes counted during an access parameter measurement.

a_ℓ = Access Denial outcome counter.

A_ℓ = Total number of Access Denials counted during an access parameter measurement.

a_m = Incorrect Access outcome counter.

A_m = Total number of Incorrect Access outcomes counted during an access parameter measurement.

$t(a)$ = Time a particular access attempt starts.

$t(a_s)$ = Time Successful Access is attained on a particular access attempt.

$w(a_s)$ = Value of access time measured on a particular successful access attempt.

r_a = Access responsibility indicator.

Figure 3.16 (sheet 2). Access parameter definition flowchart.

outcome variables should be examined; and provides a procedure for calculating values for the selected access performance parameters, given a measured sample population of access attempts. Note that the flowchart depicts a flow of logical decisions, from an a posteriori viewpoint, rather than a flow of time. The flowchart specifies a data reduction procedure, which would normally be performed off-line, rather than an on-line measurement method.

To simplify description, we assume the flowchart is implemented in a computer program. The program, then, begins by initializing variables required to compute the sample proportions of interest. For each access attempt in the measured sample population, the program executes a series of logical tests to determine which of four outcomes the attempt encountered: Successful Access, Incorrect Access, Access Denial (System Blocking), or User Blocking. The first test determines whether the access attempt resulted in Access Failure due to blocking. Blocking is defined to have occurred if the first bit of user information is not transferred across the functional interface separating the source user and the system within $3W_N(A_s)$, the maximum access time; i.e., 1B=0 in Figure 3.7.

The processing of nonblocked access attempts differs in message-oriented and circuit-oriented systems. In the former case, the nonoriginating user commitment variable (COMT) is not relevant, and the access attempt is immediately placed in the Successful Access category. In the latter case, the program tests COMT, to determine whether the intended nonoriginating user was in fact committed to the transaction during the access attempt. If so, the attempt is placed in the Successful Access category; otherwise the attempt is placed in the Incorrect Access category. Each time an attempt is placed in the Successful Access category, the number of successful access attempts (a_s) is incremented by one; and a single value of Access Time $t(a_s)-t(a)$ is computed. Each time an attempt is placed in the Incorrect Access category, the number of incorrect access attempts (a_m) is incremented by one.

In the case where Access Failure due to blocking does occur, the program examines the outcome variables SBS, UBS, and r_a to determine whether user blocking or system blocking was "responsible." If the system issued a blocking signal (i.e., SBS=1), the access attempt is immediately placed in the Access Denial (System Blocking) category. If the user issued a blocking signal, (i.e., UBS=1), the access attempt is immediately placed in the User Blocking category. Otherwise, the program determines the fate of the access attempt using the access Responsibility Indicator r_a (described more fully in Sec. 5). If r_a=1, the failure is attributed to the user, and conversely. Each time an access attempt is placed in the Access Denial category, the number of Access Denials (a_ℓ) is incremented by one. The total number of access attempts counted in the parameter calculation (a') is incremented by one each time an access attempt is placed in the Successful Access, Incorrect Access, or Access Denial category. When a' reaches a predetermined number A', the program proceeds to calculate values for the three access parameters, based on counter values a_ℓ, a_m, and a' and the set of individual access time values $w(a_s)$. Sheet 2 of Figure 3.16 presents equation definitions for the three selected parameters, together with various supplementary notes.

3.4.2.2. User Information Transfer Parameters

The user information transfer process comprises three primary functions: bit transfer, block transfer, and message transfer. The bit and block transfer functions were considered together in selecting performance parameters, since their sample spaces include the same six outcomes. For conciseness, we refer in the following only to the block transfer parameters; but it is emphasized that an exactly analogous set of parameters was selected to describe performance (separately) for the bit transfer function. In essence, we view each bit transfer parameter as a "special case" of the corresponding block transfer parameter, i.e., the case in which the user information "block" contains only one bit. The general symbols b and B refer to either the bit transfer or the block transfer function.

In considering the block transfer function, it was again concluded that the conditional event probabilities $P(i|I)$ and the waiting times to specified outcome $W(i)$ were the most significant parameter classes, for the reasons discussed in Section 3.4.2.1; but in this case, different subsets were chosen for the probability parameters, to relate each outcome to the corresponding number of possible occurrences. Given the decision to exclude b_f outcomes from the sample population in calculating parameters, the appropriate subsets (I) for each outcome (i) are the following:

Outcome (i)	Subset (I)
B_s	$(B'-B_x)$
B_ℓ	$(B'-B_x)$
B_x	$(B'-B_\ell)$
B_m	$(B'-B_\ell-B_x)$
B_e	$(B'-B_\ell-B_x-B_m) = B_s+B_e$

These subsets normalize the maximum value of each outcome probability to 1. Successful Block Transfers and Lost Blocks are expressed as a proportion of total blocks transmitted $(B'-B_x)$; Extra Blocks are expressed as a proportion of total blocks received $(B'-B_\ell)$; Misdelivered Blocks are expressed as a proportion of total blocks transferred between the source and destination in question $(B'-B_\ell-B_x)$; and Incorrect Blocks are expressed as a proportion of total blocks transferred between the source and destination in question, excluding misdelivered blocks $(B'-B_\ell-B_x-B_m)$. The latter subset can be expressed more simply as (B_s+B_e), the "successfully delivered" blocks. It would be possible to identify content errors in misdelivered blocks, but this would complicate the measurement process.

These choices of parameter classes and subsets reduced the number of possible block transfer performance parameters to ten: one conditional probability and one waiting time associated with each of the five outcomes included in b'. Five of these parameters were ultimately selected for inclusion in the proposed Federal Standard, as indicated in Figure 3.17: one parameter

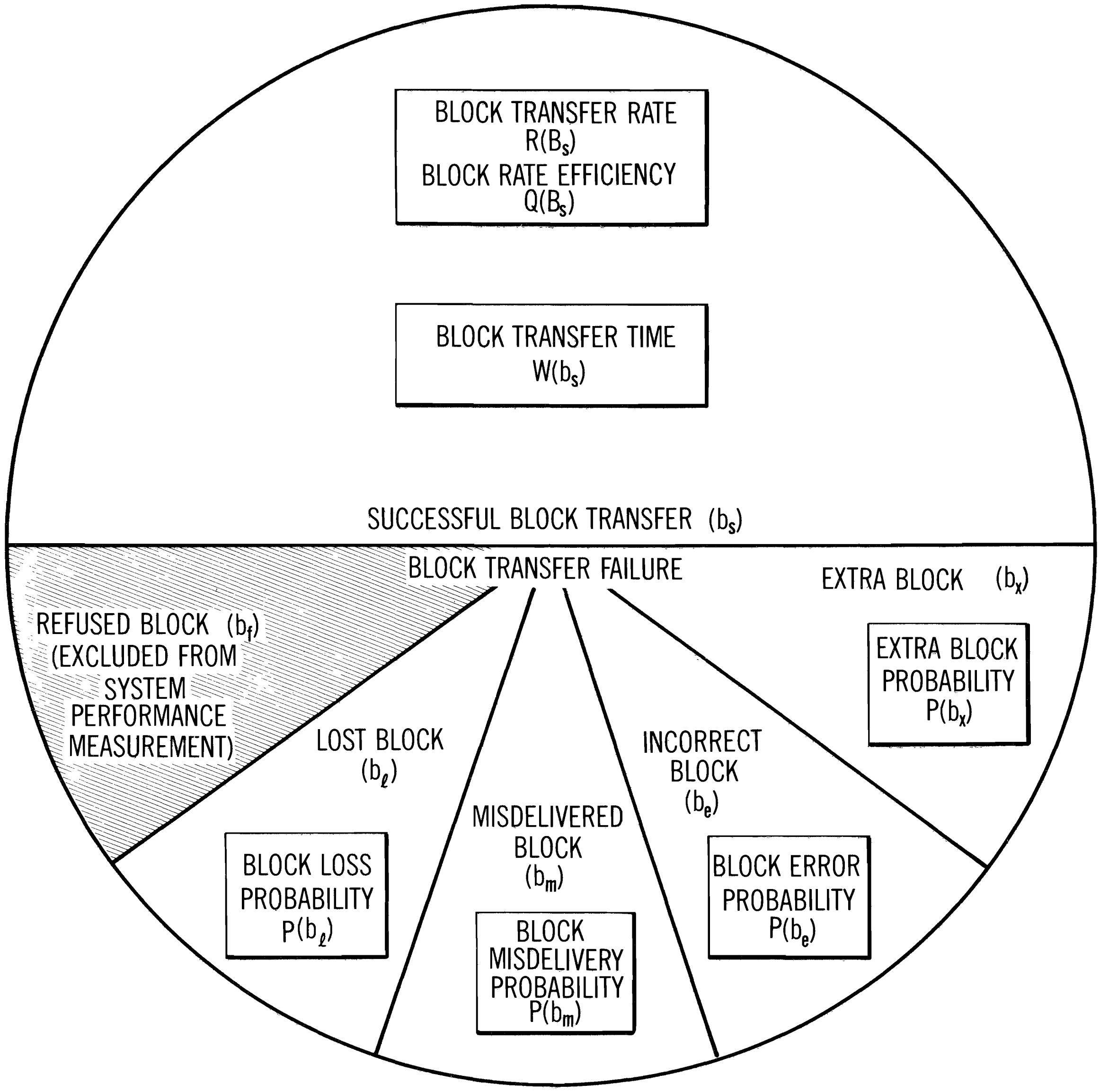


Figure 3.17. Selected user information transfer parameters.

associated with the performance criterion efficiency, three associated with the criterion accuracy, and one associated with the criterion reliability[5]. The four waiting times associated with the failure outcomes and the Successful Block Transfer Probability $P(b_s)$ were excluded for the reasons discussed (in connection with the access function) in Section 3.4.2.1. Narrative/symbolic definitions for the selected block transfer parameters are provided below.

<u>Block Transfer Time</u> $W(B_s)$ - Average value of elapsed time between the start of a block transfer attempt $t(b)$ and Successful Block Transfer $t(b_s)$. Elapsed time values are calculated only on block transfer attempts that result in Successful Block Transfer.

<u>Block Error Probability</u> $P(B_e)$ - Ratio of total Incorrect Blocks (B_e) to total successfully delivered blocks (B_s+B_e).

<u>Block Misdelivery Probability</u> $P(B_m)$ - Ratio of total Misdelivered Blocks (B_m) to total transferred blocks $(B'-B_\ell-B_x)$ in the reduced sample population B'.

<u>Block Loss Probability</u> $P(B_\ell)$ - Ratio of total Lost Blocks (B_ℓ) to total transmitted blocks $(B'-B_x)$ in the reduced sample population B'.

<u>Extra Block Probability</u> $P(B_x)$ - Ratio of total Extra Blocks (B_x) to total received blocks $(B'-B_\ell)$ in the reduced sample population B'.

The maximum block transfer time is defined as three times the specified (nominal) value of the parameter Block Transfer Time, i.e., $3W_N(b2_s)$. The rationale for the choice of the constant 3 as a performance time multiplier was presented in Section 3.4.2.1.

All of the above definitions apply equally to the bit transfer parameters, with two exceptions:

1. The maximum bit transfer time is defined to be equal to the maximum block transfer time, $3W_N(b2_s)$.

---

[5]The selected message transfer parameters (described in the following) are also identified in Figure 3.17.

2. Bit Transfer Time values are calculated only for the first and last bits of each block, since these are the only bits for which SU output and DU input times are always available.

The maximum bit transfer time cannot be less than the maximum block transfer time, since all bits in a transferred block may be delivered to the destination user simultaneously. The maximum bit transfer time cannot be more than the maximum block transfer time, since this would imply that a bit could arrive later than the associated block. The sample size used in computing the bit transfer and block transfer parameters is the same as that used in computing the message transfer parameters, i.e., B1' bits. The constant B1' excludes user Bit Transfer Failures ($B1_f$).

In the case of the third user information transfer function, message transfer, it was concluded that two of the four repeated-trial parameter classes defined in Section 3.4.1 were of significance: time rates $R(B_s)$ and rate efficiencies $Q(B_s)$. Parameters based on the $f(G_i)$ distribution, such as $P(m,n)$, were eliminated on the basis of their relative complexity. The proportion parameters $\hat{P}(i)$ were not considered separately as multi-trial parameters since they are used in calculating relative frequency estimates for the corresponding single-event probabilities, $P(i)$.

The time rate and rate efficiency parameter classes were applied to both the Successful Bit Transfer ($B1_s$) and the Successful Block Transfer ($B2_s$) outcomes, to define four primary parameters, all associated with the performance criterion efficiency. These parameters are shown in the sample space of Figure 3.17; and are defined in narrative/symbolic form (again using the term "block" in the general sense, to refer to either a block or a bit) as follows:

<u>Block Transfer Rate</u> $R(b_s)$ - Total number of Successful Block Transfers (B_s) counted during a performance measurement period, divided by the duration of the period w(b3*). The duration w(b3*) is measured in User Information Transfer Time.

Block Rate Efficiency $Q(b_s)$ - Ratio of the product of the Block Transfer Rate $R(b_s)$ and the Average Block Length (n) to the Signalling Rate of the communication service, R(max).

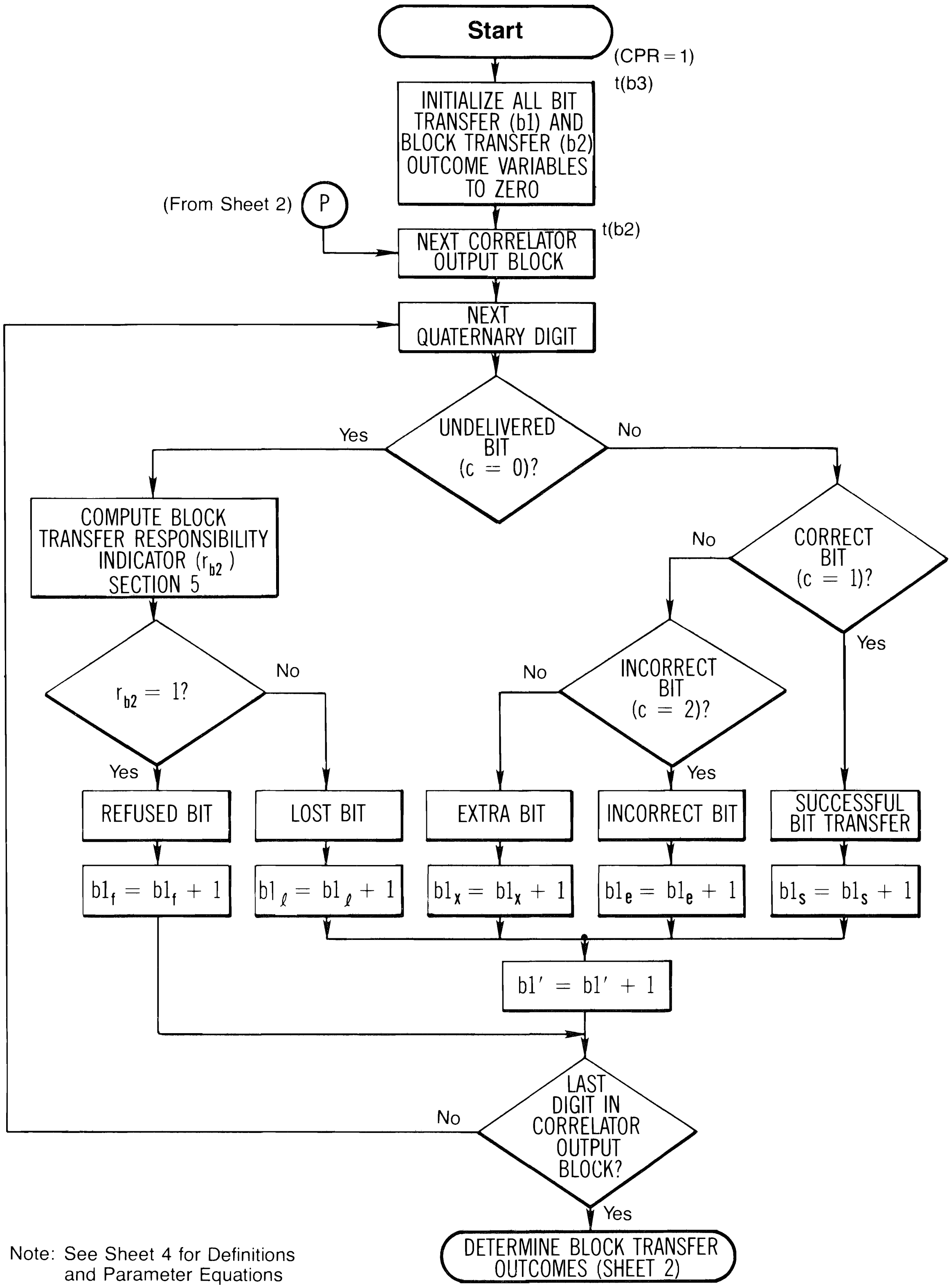
R_{max} is a constant characteristic of the communication service interconnecting the users, as described in Section 2.5.

One additional parameter, Outage Probability, was selected to express message transfer performance relative to the accuracy and reliability criteria. Although this parameter could be defined as a primary parameter, its definition is deferred to Section 4 since it is more readily described in conjunction with the secondary parameters.

Figure 3.18 is a parameter definition flowchart for the bit transfer, block transfer, and message transfer functions. The flowchart complements the bit transfer and block transfer outcome tables of Figures 3.10 and 3.11 by defining a logical sequence in which the outcome variables should be examined; and provides a procedure for calculating values for the selected user information transfer parameters, given a suitable collection of correlator output records of the form shown in Figure 3.8.

The flowchart is comprised of four sheets. Sheet 1 provides a procedure for distinguishing all bit transfer outcomes except Misdelivered Bit, based on (1) the correlator output digits, c, produced by comparing the source user Output record with the Input record of the intended destination user; and (2) the block transfer Responsibility Indicator (r_{b2}). Sheet 2 provides a procedure for distinguishing all block transfer outcomes except Misdelivered Block, based on the bit transfer outcomes b_i within each correlator output block.

Sheet 3 provides a procedure for distinguishing Misdelivered Bit and Misdelivered Block outcomes from the Extra Bit and Extra Block outcomes identified in Sheets 1 and 2. This is accomplished by comparing all Extra Blocks discovered in the first comparison with the records of source user outputs to other destinations. "Successful" outcomes of this secondary comparison (c=1 or 2) identify information unintentionally transferred



Note: See Sheet 4 for Definitions and Parameter Equations

Figure 3.18 (sheet 1). User information transfer parameter definition flowchart.

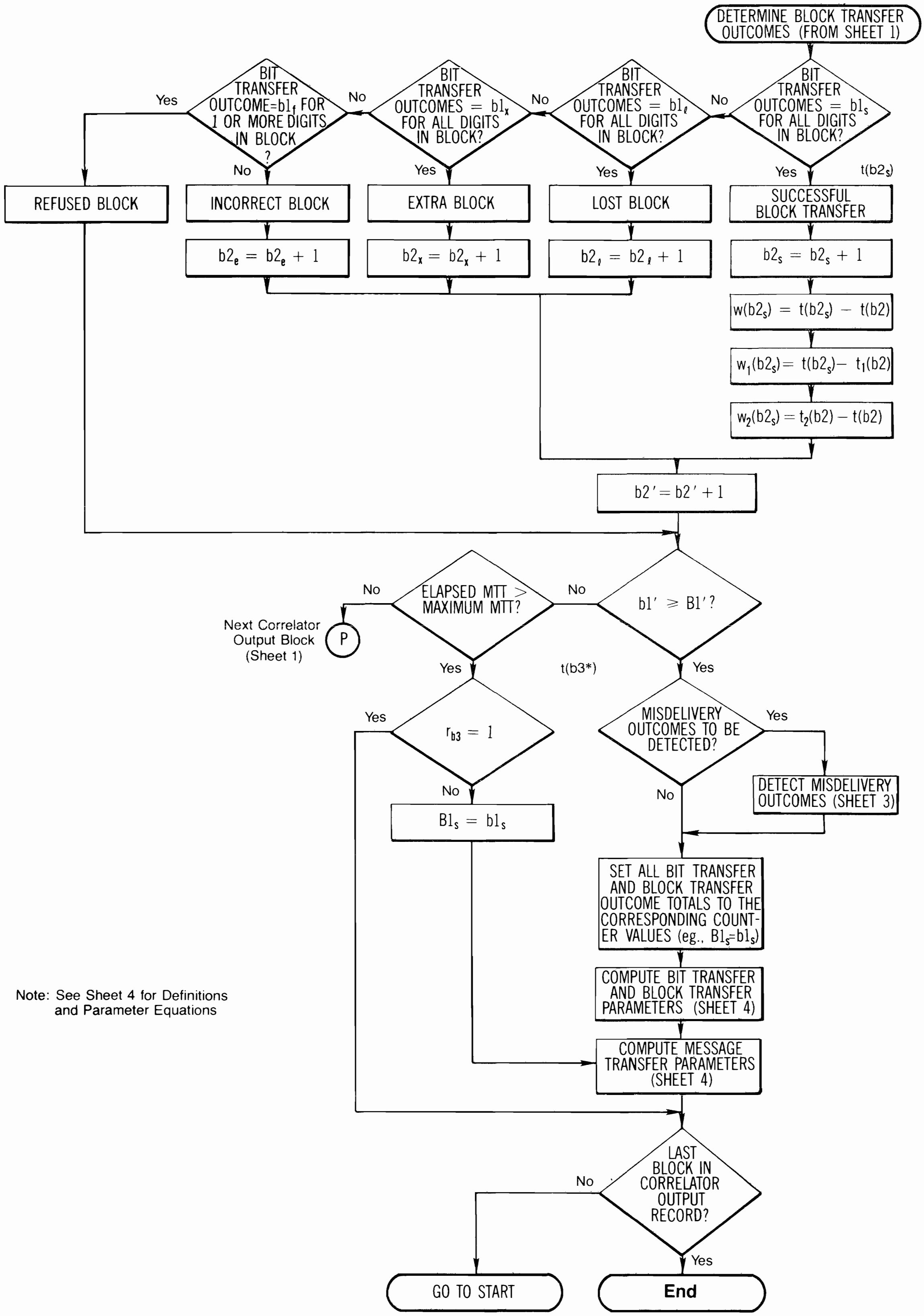


Figure 3.18 (sheet 2). User information transfer parameter definition flowchart.

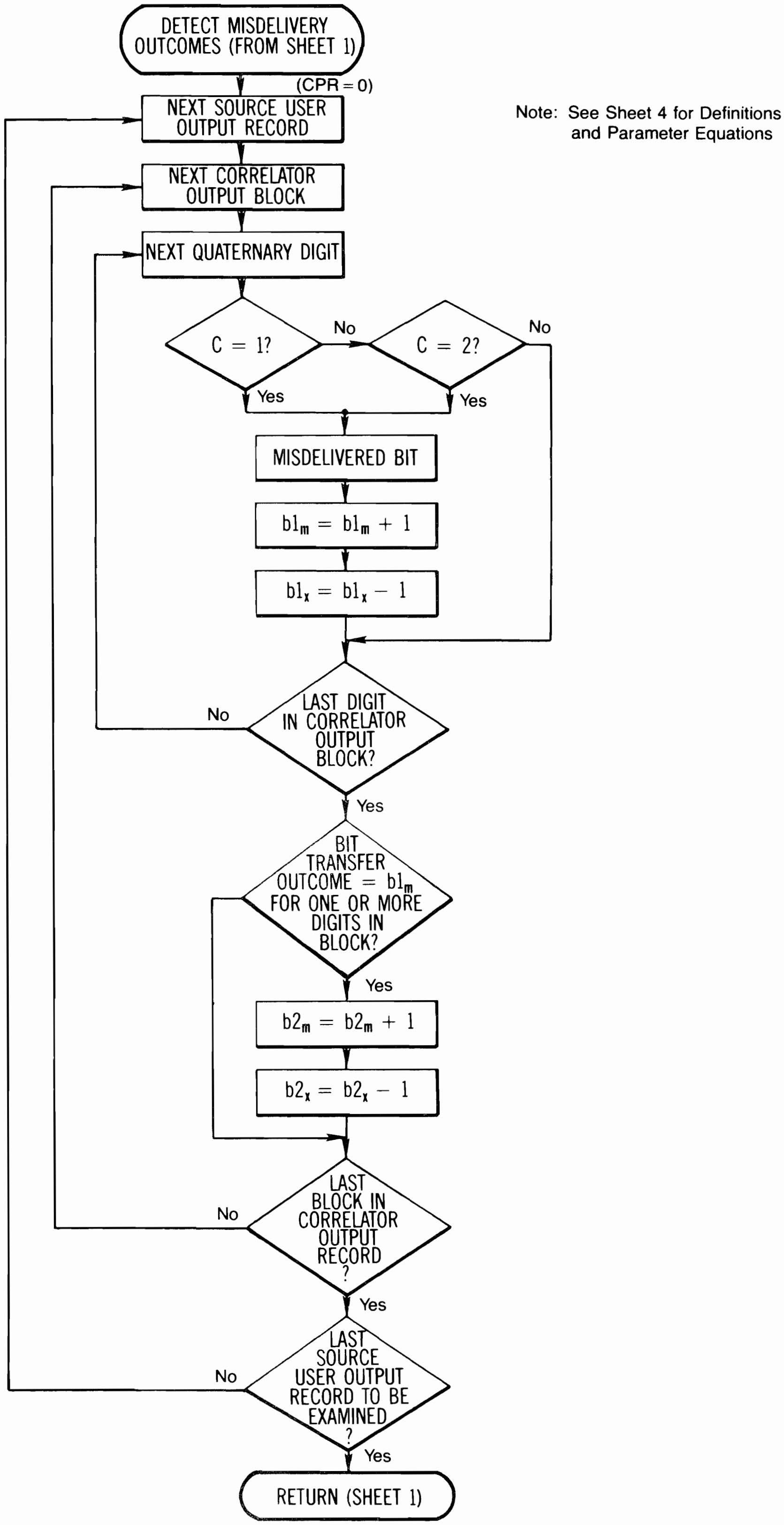


Figure 3.18 (sheet 3). User information transfer parameter definition flowchart.

BIT TRANSFER PARAMETERS

1. Bit Transfer Time = $W(bl_s) = \frac{1}{2(B2_s)} \sum_{b2_s=1}^{B2_s} w_1(b2_s) + \frac{1}{2(B2_s)} \sum_{b2_s=1}^{B2_s} w_2(b2_s)$

2. Bit Loss Probability = $P(bl_\ell) = Bl_\ell/(Bl'-Bl_x)$

3. Bit Misdelivery Probability = $P(bl_m) = Bl_m/(Bl'-Bl_\ell-Bl_x)$

4. Bit Error Probability = $P(bl_e) = Bl_e/(Bl_s+Bl_e)$

5. Extra Bit Probability = $P(bl_x) = Bl_x/(Bl'-Bl_\ell)$

BLOCK TRANSFER PARAMETERS

1. Block Transfer Time = $W(b2_s) = \frac{1}{B2_s} \sum_{b2_s=1}^{B2_s} w(b2_s)$

2. Block Loss Probability = $P(b2_\ell) = B2_\ell/(B2'-B2_x)$

3. Block Misdelivery Probability = $P(b2_m) = B2_m/(B2'-B2_\ell-B2_x)$

4. Block Error Probability = $P(b2_e) = B2_e/(B2_s+B2_e)$

5. Extra Block Probability = $P(b2_x) = B2_x/(B2'-B2_\ell)$

MESSAGE TRANSFER PARAMETERS

1. Bit Transfer Rate = $R(bl_s) = \frac{Bl_s}{w(b3^*)}$

2. Block Transfer Rate = $R(b2_s) = \frac{B2_s}{w(b3^*)}$

3. Bit Rate Efficiency = $Q(bl_s) = \frac{R(bl_s)}{R_{max}}$

4. Block Rate Efficiency = $Q(b2_s) = \frac{R(b2_s) \cdot n}{R_{max}}$

DEFINITION OF CONSTANTS

Bl' = Total number of bit transfer outcomes to be included in an individual user information transfer performance measurement.

R_{max} = Signalling rate (bits per second).

n = Average block length.

DEFINITION OF VARIABLES

$bl'(b2')$ = Bit (block) transfer outcome counter. The count excludes attempts that fail due to user non-performance.

$bl_s(b2_s)$ = Successful Bit (Block) Transfer outcome counter.

$Bl_s(B2_s)$ = Total number of Successful Bit (Block) Transfer outcomes counted during a UIT performance measurement.

bl_f = Refused Bit outcome counter.

Bl_f = Total number of Refused Bit outcomes counted during a UIT performance measurement.

$bl_\ell(b2_\ell)$ = Lost Bit (Block) outcome counter.

$Bl_\ell(B2_\ell)$ = Total number of Lost Bit (Block) outcomes counted during a UIT performance measurement.

$bl_m(b2_m)$ = Misdelivered Bit (Block) outcome counter.

$Bl_m(B2_m)$ = Total number of Misdelivered Bit (Block) outcomes counted during a UIT performance measureme

$bl_e(b2_e)$ = Incorrect Bit (Block) outcome counter.

$Bl_e(B2_e)$ = Total number of Incorrect Bit (Block) outcomes counted during a UIT performance measurement.

$bl_x(b2_x)$ = Extra Bit (Block) outcome counter.

$Bl_x(B2_x)$ = Total number of Extra Bit (Block) outcomes counted during a UIT performance measurement.

$t(b3)$ = UIT time a particular UIT performance measurement starts.

$t(b3^*)$ = UIT time a particular UIT performance measurement ends.

$t(b2)$ = Time a particular block transfer attempts starts.

$t(b2^*)$ = Time a particular block transfer attempt ends.

c = Data correlator output.

$w(b2_s)$ = Value of block transfer time measured on a particular successful block transfer attempt.

$w_1(b2_s)$ = Value of bit transfer time measured on the last bit of a particular successful block transfer attempt.

$w_2(b2_s)$ = Value of bit transfer time measured on the first bit of a particular successful block transfer attempt.

CPR = Comparison variable.

r_{b2} = Block transfer responsibility indicator.

r_{b3} = Bit transfer responsibility indicator.

Figure 3.18 (sheet 4). User information transfer parameter definition flowchart.

between the source/destination pair considered in the initial comparison. As was discussed previously, the additional data reduction effort required to identify misdelivery outcomes may not be justified in all situations. Nevertheless, the misdelivery probabilities can be very useful in applications where privacy and security are important to the end users; these parameters provide a specific measure of the protection a system affords against inadvertent user information disclosure. They are specifically applicable to the problem of data communications audit in an ADP environment (e.g., see FitzGerald, 1976).

Sheet 4 presents equation definitions for the fourteen primary user information transfer parameters, together with various supplementary notes.

3.4.2.3. Disengagement Parameters

The disengagement sample space is identical to the access sample space, except that "incorrect disengagement" is not distinguished as a separate outcome since each disengagement function applies to a single user interface. The same parameter classes selected to represent access performance were also selected to represent disengagement; i.e., the conditional event probabilities $P(i|I)$, for the subset d' of trials excluding User Blocking; and the corresponding waiting times to specified outcome, $W(i)$.

These choices reduced the set of possible disengagement parameters to four: one probability and one waiting time associated with each of the two remaining outcomes, d_s and d_ℓ. Two of these parameters were ultimately selected for inclusion in the proposed Federal Standard (Fig. 3.19): one associated with the performance criterion efficiency, and one addressing both the accuracy and reliability criteria. Disengagement Denial Time and Successful Disengagement Probability were excluded for the reasons discussed (in connection with the access function) in Section 3.4.2.1. Narrative/symbolic definitions for the selected disengagement parameters are provided below.

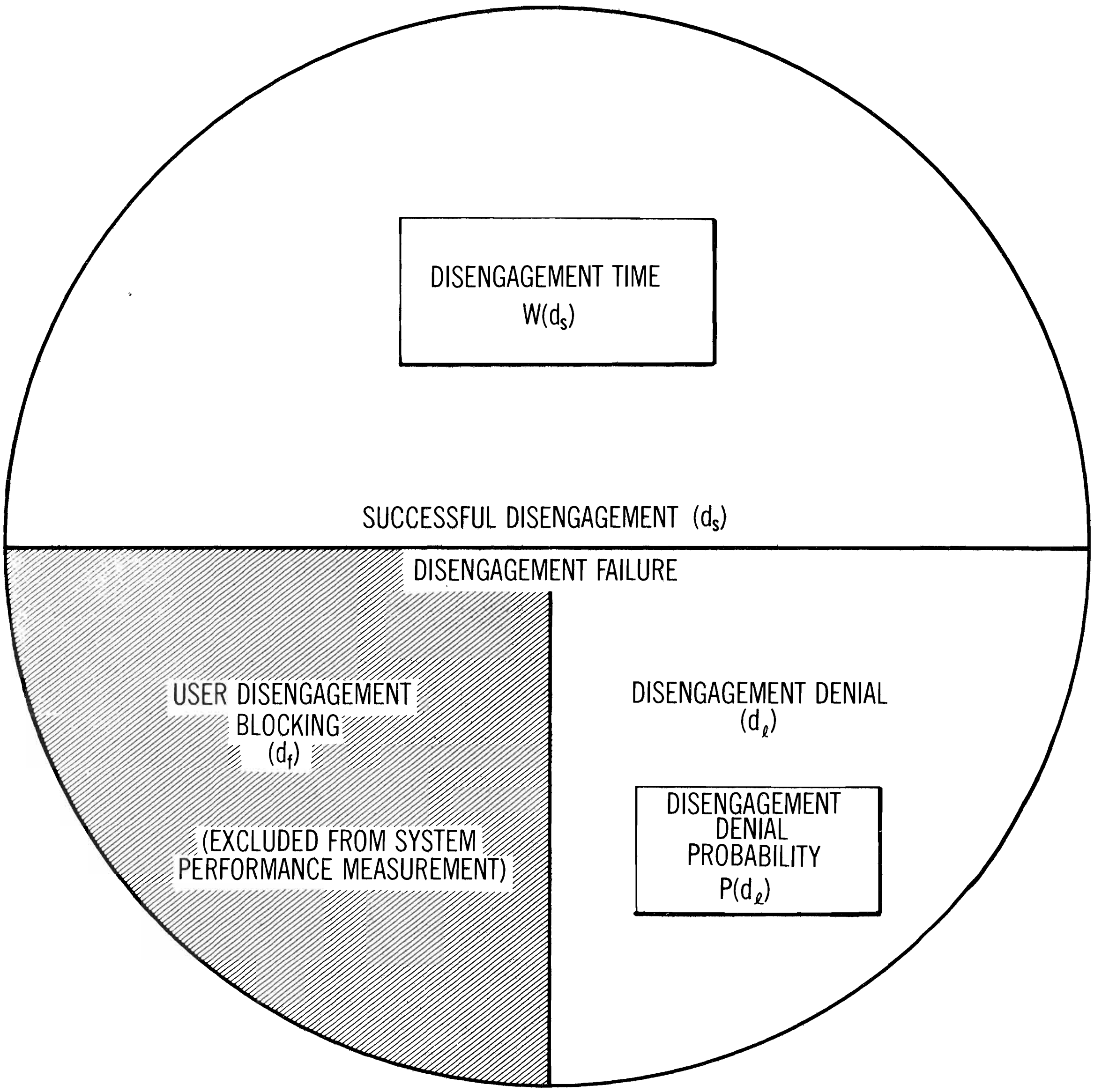


Figure 3.19. Selected disengagement parameters.

Disengagement Time $W(D_s)$ - Average value of elapsed time between the start of a disengagement attempt t(d) and Successful Disengagement $t(d_s)$. Elapsed time values are calculated only on disengagement attempts that result in Successful Disengagement.

Disengagement Denial Probability $P(D_\ell)$ - Ratio of total disengagement attempts that result in Disengagement Denial (D_ℓ) to total disengagement attempts included in the reduced sample population (D').

Figure 3.20 is a parameter definition flowchart for the disengagement function. The flowchart complements the disengagement outcome table of Figure 3.13 by defining a logical sequence in which the outcome variables should be examined; and provides a procedure for calculating values for the selected disengagement parameters, given a measured sample population of disengagement attempts. The flowchart is basically a simplification of the access parameter definition flowchart of Figure 3.16, excluding the tests associated with issuance of blocking signals and with incorrect performance. Sheet 2 of Figure 3.20 presents equation definitions for the selected disengagement parameters, together with various supplementary notes.

3.4.3. Summary

Figure 3.21 summarizes the 19 primary performance parameters selected for inclusion in the proposed Federal Standard. The defined parameters include three access parameters; five bit transfer parameters; five block transfer parameters; four message transfer parameters; and two disengagement parameters. The shaded areas in Figure 3.21 indicate the performance parameters that remain to be defined. These include the secondary parameters (including Outage Probability), and the ancillary parameters. The secondary parameters are defined in Section 4; the ancillary parameters are defined in Section 5.

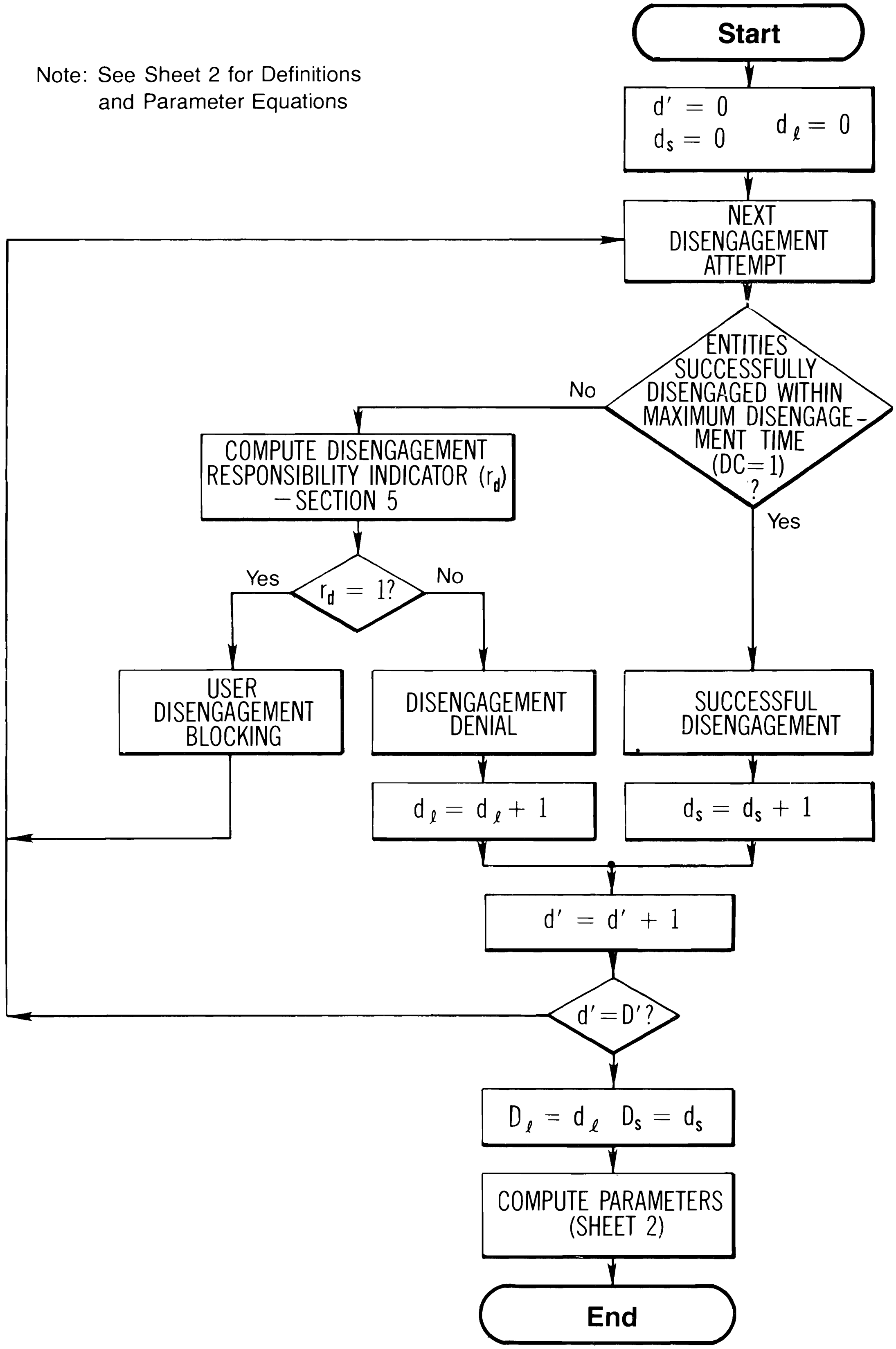


Figure 3.20 (sheet 1). Disengagement parameter definition flowchart.

DISENGAGEMENT PARAMETERS

1. Disengagement Time = $W(d_s) = \frac{1}{D_s} \sum_{d_s=1}^{D_s} w(d_s)$

2. Disengagement Denial Probability = $P(d_\ell) = D_\ell/D'$

3. DEFINITION OF CONSTANTS

   D' = Total number of disengagement attempts counted during a disengagement parameter measurement.

DEFINITION OF VARIABLES

d' = Disengagement attempt counter. The count excludes attempts that fail due to user disengagement blocking.

d_s = Successful disengagement outcome counter.

D_s = Total number of successful disengagement outcomes counted during a disengagement parameter measurement.

d_ℓ = Disengagement denial outcome counter.

D_ℓ = Total number of disengagement denials counted during a disengagement parameter measurement.

$t(d)$ = Time a particular disengagement attempt starts.

$t(d_s)$ = Time successful disengagement is attained on a particular disengagement attempt.

$w(d_s)$ = Value of disengagement time measured on a particular successful disengagement attempt.

r_d = Disengagement responsibility indicator.

Figure 30 (sheet 2). Disengagement parameter definition flowchart.

FUNCTION	PERFORMANCE CRITERION		
	EFFICIENCY	ACCURACY	RELIABILITY
ACCESS (a)	• ACCESS TIME $W(a_s)$	• INCORRECT ACCESS PROBABILITY $P(a_m)$	• ACCESS DENIAL PROBABILITY $P(a_\ell)$
BIT TRANSFER (b1)	• BIT TRANSFER TIME $W(b1_s)$	BIT MISDELIVERY PROBABILITY $P(b1_m)$ BIT ERROR PROBABILITY $P(b1_e)$ EXTRA BIT PROBABILITY $P(b1_x)$	• BIT LOSS PROBABILITY $P(b1_\ell)$
BLOCK TRANSFER (b2)	• BLOCK TRANSFER TIME $W(b2_s)$	• BLOCK MISDELIVERY PROBABILITY $P(b2_m)$ • BLOCK ERROR PROBABILITY $P(b2_e)$ • EXTRA BLOCK PROBABILITY $P(b2_x)$	• BLOCK LOSS PROBABILITY $P(b2_\ell)$
MESSAGE TRANSFER (b3)	• BIT TRANSFER RATE $R(b1_s)$ • BIT RATE EFFICIENCY $Q(b1_s)$ • BLOCK TRANSFER RATE $R(b2_s)$ • BLOCK RATE EFFICIENCY $Q(b2_s)$	• OUTAGE PROBABILITY	
DISENGAGEMENT (d)	• DISENGAGEMENT TIME $W(d_s)$	• DISENGAGEMENT DENIAL PROBABILITY $P(d_\ell)$	

ANCILLARY PARAMETERS

SECONDARY PARAMETERS

Legend
- Primary Parameters
- Secondary Parameters
- Ancillary Parameters

Figure 3.21. Summary of primary parameters.

4. SECONDARY PARAMETERS

4.1. Introduction

A frequently quoted definition of reliability is that of the Advisory Group on the Reliability of Electronic Equipment (AGREE, 1957):

> "Reliability is the probability of performing without failure a specified function under given conditions for a specified period of time."

ANSI's character-oriented data communication performance standard X3.44 (ANSI, 1974) defines the availability of an information path as follows:

> "The portion of a selected time interval during which the Information Path is capable of performing its assigned data communications function. Availability is expressed as a percentage."

These definitions clearly reflect a more general, more macroscopic view of performance than was used in defining the primary performance parameters. The traditional reliability definition does not distinguish between incorrect performance and non-performance, as was done in Section 3; instead, the two outcome categories are combined under the more general concept of performance failure. The quoted availability definition reflects an even more macroscopic view: both successful and unsuccessful performance periods must be observed to determine "the portion of a selected time interval during which the Information Path is capable of performing its assigned data communications function."

Analysis of the surveyed sources indicated considerable user interest in parameters describing "overall" telecommunication system performance from this more macroscopic point of view. The Subcommittee therefore undertook to define a set of "secondary" performance parameters to meet this expressed need. This section summarizes the technical considerations which influenced the choice of secondary parameters; and defines the selected parameters on the basis of a simple two-state (Markov) outage model.

The section is divided into three major subsections. The first subsection defines certain traditional reliability and availability concepts which are essential to the subsequent discussion. The second subsection describes the functions and outcomes considered in defining the key concept of outage. The third subsection formally defines the secondary functions, outcomes, and parameters, using the same descriptive tools employed in Section 3: state diagrams, sample spaces, and parameter definition flowcharts.

4.2. Traditional Reliability and Availability Concepts

The most fundamental concept in traditional reliability analysis is the reliability function r(t). This function is essentially a mathematical formulation of the AGREE definition quoted above: it expresses the probability that a specified function will be performed without failure, as a function of performance time, t. Reliability theorists commonly represent the reliability function as an exponential function, for mathematical simplicity:

$$r(t) = e^{-\lambda t}.$$

The parameter λ is termed the failure rate of the system; it is defined as the expected number of system failures per unit time. The expected value of the probability distribution associated with r(t) is termed the mean time between failures (MTBF); for the exponential reliability function it is the reciprocal of the failure rate, i.e.,

$$\text{MTBF} = 1/\lambda \quad (\lambda \text{ constant}).$$

Essentially identical concepts are used to describe the "maintainability" of a system; i.e., there is a maintainability function, often represented as an exponential $m(t)=e^{-\mu t}$; the parameter μ is the repair rate; and its reciprocal is the mean time to repair (MTTR).

In the exponential case, availability can be simply expressed in terms of the failure rate λ, repair rate μ, and their reciprocals as follows:

$$A = \frac{\mu}{\mu + \lambda} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} .$$

The complement of availability, termed unavailability, is also sometimes used in reliability analysis; it is expressed as

$$U = 1-A = \frac{\lambda}{\lambda + \mu} = \frac{\text{MTTR}}{\text{MTTR} + \text{MTBF}} .$$

In principle, the function r(t) provides the most complete measure of traditional reliability, since it is the distribution on which the other statistics are based. Nevertheless, it is difficult to determine in practice; and estimates of λ and μ and their functions are much more frequently used in specifying reliability performance. The parameters MTBF, MTTR, and availability are particularly common. Availability is derivable from MTBF and MTTR; but the converse is not true.

4.3. Definition of Outage

The traditional reliability and availability concepts are all based on the assumption that the system in question can be characterized, at any point in time, as being in one of two distinct states: Operational Service (the "good" state); or Outage (the "bad" state). In practice, this characterization can only be made by observing system performance, relative to one or more defined functions, over a performance measurement period of nonzero duration. In order to describe telecommunication system performance from the macroscopic viewpoint, it was therefore necessary to select one or more primary functions which could be regarded as representing "overall" system performance; and to specify a measurement period over which the selected function(s) would be observed to determine the current secondary (outage) state. Three options were considered:

Option 1. Define the secondary states on the basis of user information transfer performance, as measured over the performance period of an individual message transfer function. This period represents the minimum sample size over which values for the primary UIT parameters can be determined.

Option 2. Define the secondary states on the basis of access performance, as measured over a specified number of access attempts.

Option 3. Combine (1) and (2) above, i.e., consider both UIT and access performance in defining the secondary states. This alternative was given serious consideration, since it does provide, in principle, a more comprehensive overall measurement.

Figure 4.1 shows the implications of these options from the standpoint of usage dependence[6]. Figure 4.1a depicts two independent random processes:

1. A system outage process, which represents the ability of the system to support communication between two specified users at any given time, independent of their actual need for service. The process is characterized by a failure rate λ and repair rate μ, as described above.
2. A user need process, which represents the need of the two users to communicate with each other, assuming the ability of the system to provide such service. Users are viewed as having a need to communicate whenever one has information it wishes to transfer to the other. This process is also characterized as exponential, with a constant "need arrival" rate α and "departure" rate β.

There are six possible ways an individual outage period and an individual usage (or service need) period can be interrelated in time. Four of these possibilities are relevant here (Fig. 4.1b):

Case 1 - Usage with no overlapping outage.

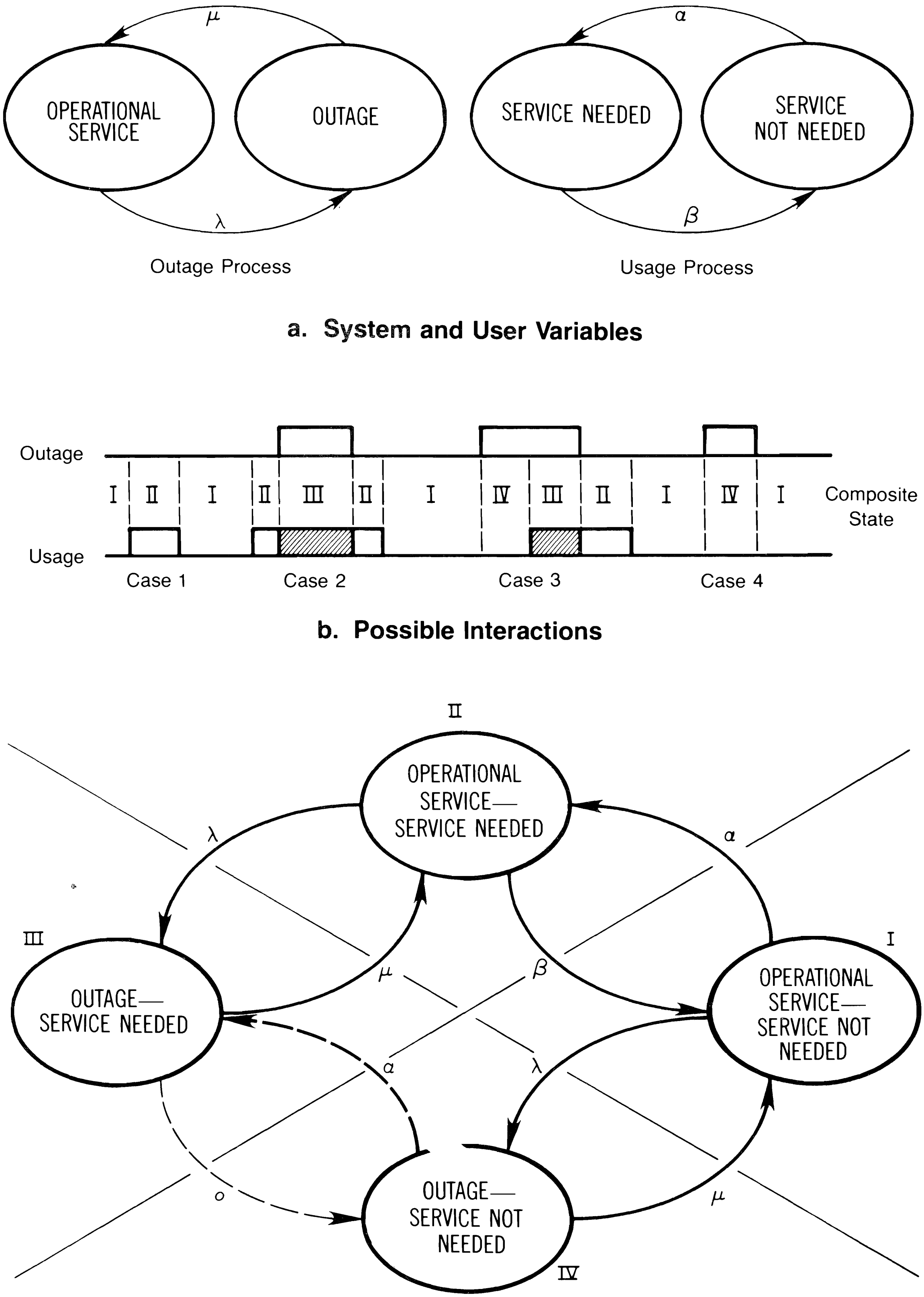
Case 2 - Outage arising and ending during a period of usage.

Case 3 - Service need arising during, and ending after, an outage.

Case 4 - Outage with no overlapping service need.

---

[6]The figure (and the subsequent reasoning) assume that the measurement time required to determine the system's secondary state is much shorter than the state's average duration.



a. System and User Variables

b. Possible Interactions

c. Composite States and Usage Dependence

Figure 4.1. Interaction between outage and usage.

The other two possibilities (service need arising and ending during an outage, and outage arising during and ending after usage) are excluded by the premise that a service need persists until the service is provided (i.e., no "lost calls").

Two significant observations can be made: first, users can be aware of system outages only during service need; and second, outages prolong any overlapping period of service need by the duration of overlap. Figure 4.1c shows the consequences of this dependence on the measurement of system outages by intermittently communicating users. Joint consideration of the outage and usage variables produces four composite states: two Operational Service states, only one of which is observable to the users; and similarly for the Outage states. Note that no transition from state III to state IV is possible, since this would imply termination of an unfulfilled need.

A typical objective of a traditional reliability measurement would be to obtain estimates for the transition rates λ and μ by recording times spent in the Operational Service and Outage states. But the users can only make observations of times spent in the two Service Needed sub-states, II and III; and these times depend (in general) on α and β as well as on λ and μ. Observations of individual waiting times in the Service Needed sub-states would not produce an unbiased estimate of system performance relative to (say) availability; and parameter values measured under different usage conditions would not be comparable. These are undesirable limitations.

The solution to this problem is to base all observations of time spent in the two Service Needed sub-states on transitions between these same states. An unbiased estimate of $1/\mu$ can be obtained by computing the average waiting time in state III, between entry from and exit to state II. We disregard cases in which state III is entered from state IV, in effect choosing a random sample of outage periods - those beginning during usage. In a similar way, an unbiased estimate $1/\lambda$ can be obtained by computing the average of the (cumulative) time spent in state III, between successive transitions III→II and II→III. This is true as long as (1) the pattern of outages is uncorrelated with the usage of the subscriber pair, i.e., $\lambda(II \rightarrow III) = \lambda(I \rightarrow IV)$; and (2) transitions IV→III are disregarded as described above.

The "cost" paid for the benefit of unbiased estimates of λ and μ is simply this: the total ("clock") time required to obtain the estimates (given a fixed confidence objective) is increased by the factor 1/u, where u is the utilization factor of the service, i.e., the ratio of Service Needed time to Total Time. As an example, if the utilization factor of a service is 0.25, the average number of (say) months required to observe a given number of outage periods would be increased by a factor of four. This is consistent with the expectation that if a random process is observed only one-fourth of the time, three-fourths of all events produced will not be included in the observed sample population.

The influence of usage dependence on the choice of primary functions to be used in defining the secondary states can now be stated:

> Option 1 restricts outage observations to transitions between the two Service Needed sub-states. The UIT functions describe performance only during user information transfer time (Fig. 3.2); and all of this time is "Service Needed" time.
>
> Options 2 and 3 both require observations of transitions between the Service Needed and Service Not Needed sub-states, since each option involves use of the access function in detecting outages.

Option 1 is preferable: it provides a method of describing service availability which is insensitive to variations in usage pattern. Intermittent measurements, made during normal operational use of a service, can thus be used to characterize its underlying outage pattern in terms of parameters such as MTBF and MTTR; and parameter values obtained under different usage conditions can be directly compared. These advantages led to the conclusion that only the UIT functions should be considered in defining the secondary performance parameters.

Given the decision to base the secondary parameters on observations of UIT performance, the next step was to select a method of defining the Operational Service and Outage states. The traditional approach to this problem (e.g., see AT&T, 1975) is to define certain key performance parameters as "supported parameters"; establish particular values for these parameters as "outage thresholds"; and then compare the measured values for all supported parameters with the associated thresholds at periodic intervals to determine the secondary state, a posteriori. A service is defined to have been in the Operational Service state (during the preceding performance measurement period) whenever the measured values for all supported parameters are better than their associated outage thresholds; and is defined to have been in the Outage state whenever the measured values for one or more supported parameters are worse than their associated thresholds.

In implementing this approach, it was decided that five UIT parameters should be defined as supported performance parameters: the four bit transfer probabilities (Bit Error Probability, Bit Misdelivery Probability, Bit Loss Probability, and Extra Bit Probability) and Bit Transfer Rate. The reason for choosing the bit-oriented parameters in preference to the block-oriented parameters was the fact that the former can be directly compared between services having different characteristic block lengths. It was decided that either an excessive number of bit transfer failures (in any of the four listed categories) or an excessively low rate of Successful Bit Transfers would nullify the value of a telecommunication service to the end users; these outcomes can occur independently in asynchronous systems.

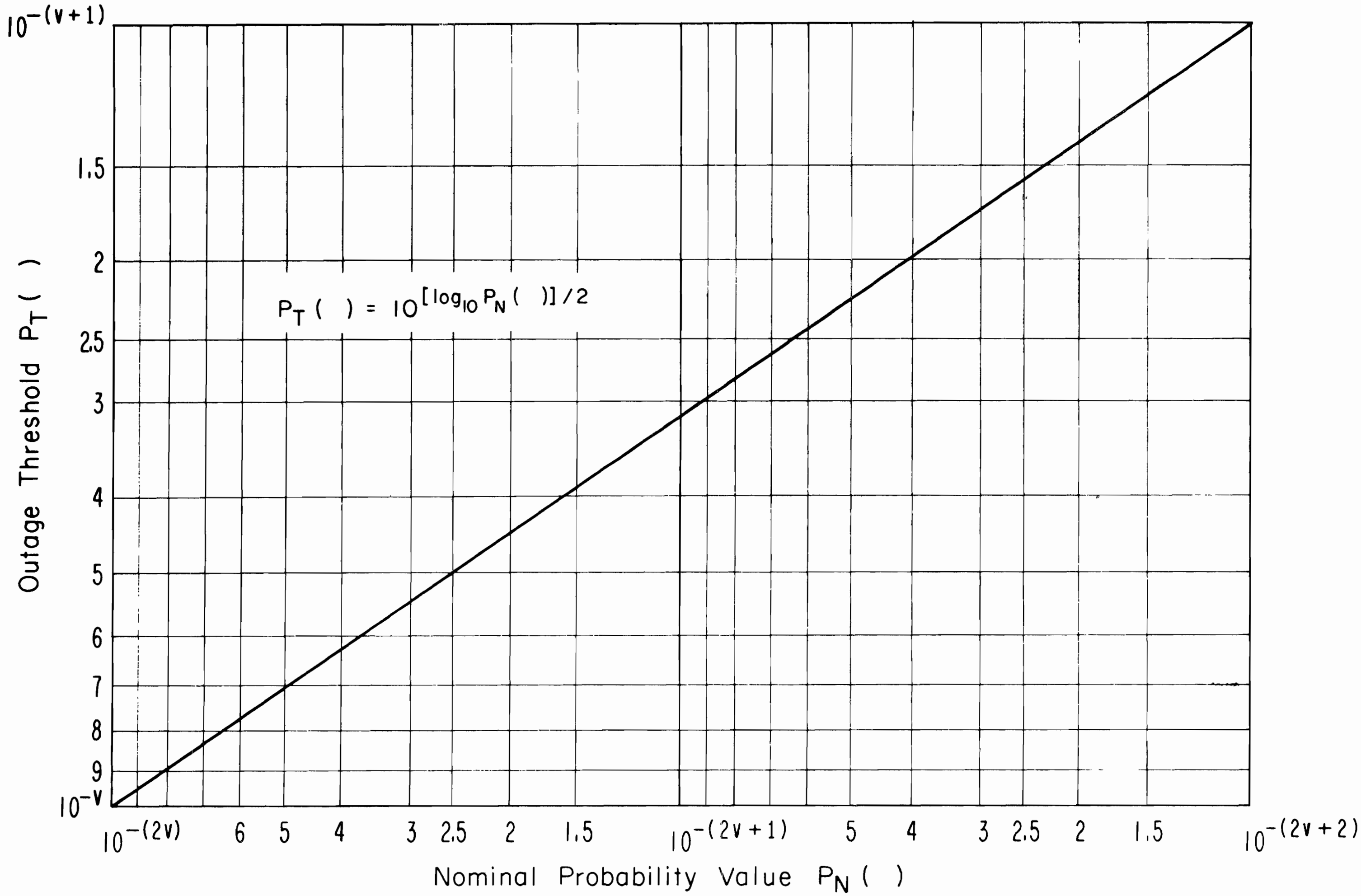
It was decided that all parameter values would be calculated over a performance measurement period equal to the message transfer time, with a maximum waiting time of $3W_N(B1')$ specified to ensure detection of "open circuit" conditions. This maximum waiting time is the UIT time required to send three "messages" at the specified bit transfer rate. The choice of the constant 3 is dictated by the requirement that the maximum performance time for the message transfer function be compatible with the maximum performance times for the bit and block transfer functions (Sec. 3.4.2.2).

Outage thresholds for the supported performance parameters were defined as a function of the corresponding nominal values (specified for the service) as follows:

1. The outage threshold for Bit Transfer Rate was defined as one-third (1/3) of the nominal Bit Transfer Rate.
2. The outage thresholds for the four bit transfer failure probabilities were defined as a function of the corresponding nominal probability values as shown in Figure 4.2. The analyst locates the specified nominal probability value on the abscissa, and then reads the corresponding outage threshold off the ordinate.

One-third nominal is the highest Bit Transfer Rate that appears reasonable as an outage threshold, given $3W_N(Bl')$ as a maximum message transfer time. This rate was judged to be more suitable than possible lower values on the basis of a subjective assessment of user expectations. It is recognized that any value in the range one-third to one-tenth could be considered to be a reasonable outage threshold.

The procedure for determining the bit transfer failure probabilities corresponds to dividing the exponent of the nominal probability value by 2. For example, for $P_N=10^{-6}$ (v=2), $P_T=10^{-3}$; for $P_N=10^{-7}$ (v=3), $P_T=3.16 \times 10^{-4}$. It was considered important to define the outage thresholds as a function of the corresponding nominal values to make the secondary parameters insensitive to differences in specified performance between service "classes." As an example, a short line-of-sight microwave link might be designed to provide a 10^{-6} bit error probability, whereas a long-distance HF digital link might be designed to provide only 10^{-2}. Selection of a fixed outage threshold of (say) 10^{-3} would be



Notes:

1. Nominal and Threshold Probability Values Are Represented in the Form $a \times 10^{-b}$, Where "b" is a Positive Integer.

2. The Scaling Variable "v" is a Positive Integer Chosen so as to Make the Nominal Probability Value $P_N(\ )$ Lie on the Abscissa.

Figure 4.2. Method of computing outage thresholds $P_T(\ )$ from nominal values $P_N(\ )$ for the probability parameters.

appropriate for the first service, but would be unreasonable for the second.

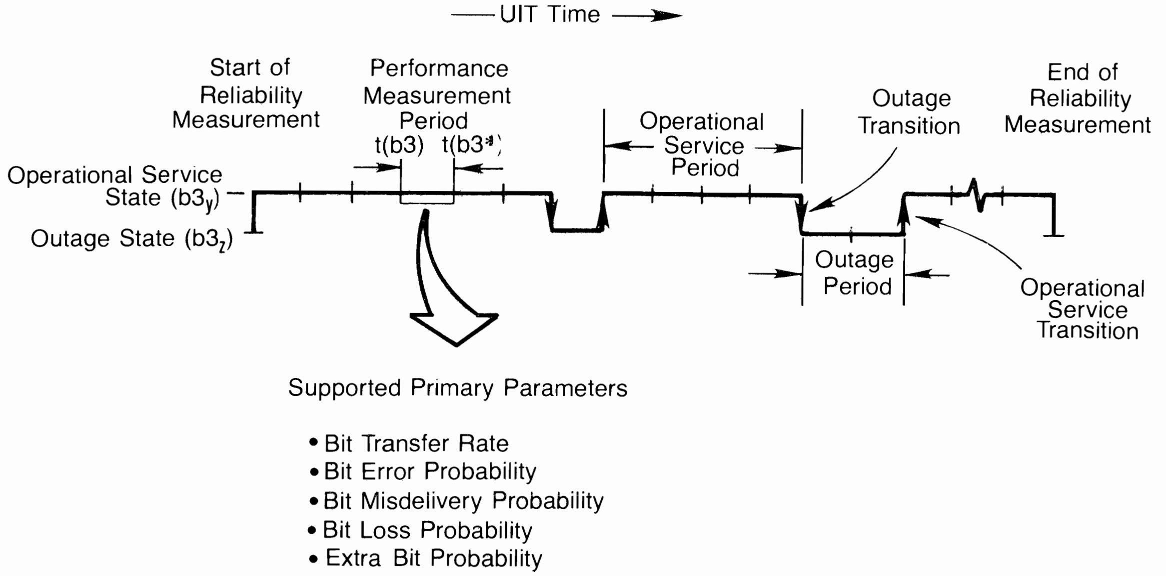
The "exponent-division" approach to defining probability thresholds is basically a compromise between two extremes: applying a high constant threshold (e.g., 10^{-1}) to all services, independent of nominal value; and defining the threshold in terms of a specified number of "excess errors" (e.g., threshold of $10^{-(n-1)}$ for nominal probability value of 10^{-n}, $n>1$). The first extreme would favor systems with relatively low specified probability values, where the second would favor systems with relatively high specified values. The constant 2 was chosen as an exponent divisor on the basis of a subjective assessment of user expectations.

4.4. Secondary Functions, Outcomes, and Parameters

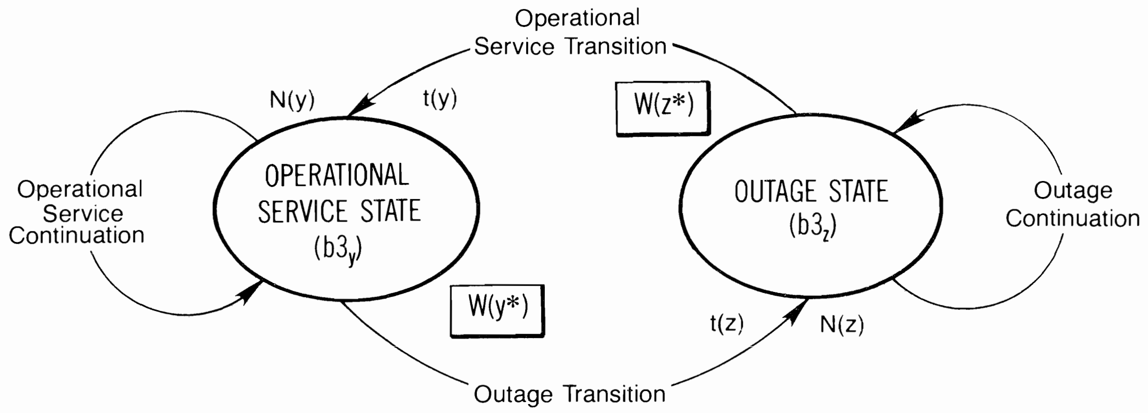
Section 2 pointed out that any description of performance refers to some functional model; and specified a discrete functional model which was used in defining the primary parameters. We now define a much simpler model to be used in defining the secondary parameters.

Figure 4.3 illustrates the selected secondary model. The "secondary state" of the telecommunication system interconnecting two users is modeled as a binary Markov process defined on the outcomes of a succession of message transfer functions. On completion of each successive function, we calculate values for the five supported performance parameters; compare the calculated values with the corresponding outage thresholds; and define the "secondary outcome" of that trial performance of the function as either Operational Service state ($b3_y$) or Outage state ($b3_z$). The term "secondary outcome" is used to emphasize the fact that these outcomes are defined on the basis of measured primary parameter values, rather than directly on the message transfer sample space.

Given two successive message transfer outcomes, there are four possible "events" in the secondary model:



a. Secondary State Sequence



b. Secondary State Diagram

Figure 4.3. Secondary model.

1. Continuation of the Operational Service state ($b3_y \rightarrow b3_y$).
2. Transition from the Operational Service state to the Outage state ($b3_y \rightarrow b3_z$).
3. Continuation of the Outage state ($b3_z \rightarrow b3_z$).
4. Transition from the Outage state to the Operational Service state ($b3_z \rightarrow b3_y$).

We define two secondary functions on the basis of events (2) and (4) above, as follows:

Service Continuation (y). This function is defined to begin on occurrence of model event (4) above; or on completion of the first message transfer function for a new service when outcome $b3_y$ occurs. The function is defined to end on occurrence of event (2) above. The function consists of maintaining the telecommunication service in the Operational Service state continuously (in UIT time), without transition to the Outage state.

Service Restoral (z). This function is defined to begin on occurrence of model event (2) above; or on completion of the first message transfer function for a new service when outcome $b3_z$ occurs. The function is defined to end on occurrence of model event (4) above. The function consists of returning the telecommunication service from the Outage state to the Operational Service state.

Consider now the possible outcomes of these complementary secondary functions. In any multi-trial experiment, there are three general outcome variables: the number of trials; the outcome of each individual trial; and the total performance time. In order to ensure a manageable sample space, it is normally necessary to define the experiment in such a way that one of these three variables is fixed. The other two can then be summarized by appropriate performance parameters. In defining the primary functions, we fixed the number of trials; and the individual trial outcomes and waiting times were thus the relevant outcome variables.

In defining the secondary parameters, we have chosen instead to focus in each case on a particular individual trial outcome - a transition between secondary states. The relevant outcome variables thus consist of the number of message transfer trials, n(y) and n(z), between complementary secondary state transitions; and the waiting times between these transitions, w(y*) and w(z*). Each of these statistics is a random variable defined on a sample space consisting of the set of positive integers[7].

These four RV's provided the basic set of options from which the secondary performance parameters were chosen. Each RV was summarized in terms of its average value, to produce four candidate parameters: the average number of message transfer trials, and the average UIT time, spent in each of the two reliability states (Fig. 4.3b). These two types of parameters differ only by a constant (the average message transfer time) in any given service; and the waiting times are clearly preferable from the standpoint of expressed user need. The two waiting time parameters W(y*) and W(z*) were therefore selected for inclusion in the Standard. Narrative/symbolic definitions for these parameters are provided below.

<u>Service Time Between Outages W(y*)</u> - Average value of elapsed user information transfer time between the start t(y) and the end t(z) of the secondary function service continuation.

<u>Outage Duration W(z*)</u> - Average value of elapsed user information transfer time between the start t(z) and the end t(y) of the secondary function service restoral.

These parameters provide essentially unbiased estimates of the traditional reliability parameters MTBF and MTTR, as defined in Section 4.3.

One additional secondary performance parameter was selected for inclusion in the standard:

---

[7] Time is again assumed to be a discrete variable.

Outage Probability $P(b3_z)$ - Ratio of total message transfer attempts resulting in the secondary outcome Outage state ($B3_z$) to total message transfer attempts included in the reduced sample population (B3').

Outage Probability was included among the selected secondary parameters for two reasons:

1. It is more readily measured than Outage Duration or Service Time Between Outages. Its sample size would typically be based on a given number of message transfer trials, whereas the latter parameters would require a sample containing many reliability state transitions.
2. It provides more meaningful information in the case where the rate of variation in system performance is high compared to the rate at which the secondary state can be determined (the "message" transfer rate).

Outage probability is essentially a discrete (sampled) measure of unavailability. There may be performance assessment situations in which only Outage Probability is justified as a secondary performance measure.

Figure 4.4 is a parameter definition flowchart for the secondary parameters. To simplify description, we again assume the flowchart is implemented in a computer program. Input to the program is a succession of secondary message transfer outcomes, each comprising a set of five measured primary parameter values - one value for each supported parameter. These values would be determined using the procedure defined in Figure 3.18.

In processing each secondary outcome, the program compares the measured parameter values with the corresponding outage thresholds; and thereby derives a single (binary) value for the secondary state variable T(b3). If all measured parameter values are better than their associated thresholds, T(b3) is set to zero, to indicate the Operational Service state. If one or more measured parameter values is worse than its threshold, T(b3) is set to 1, to indicate the Outage state; and the Outage State counter ($b3_z$) is incremented.

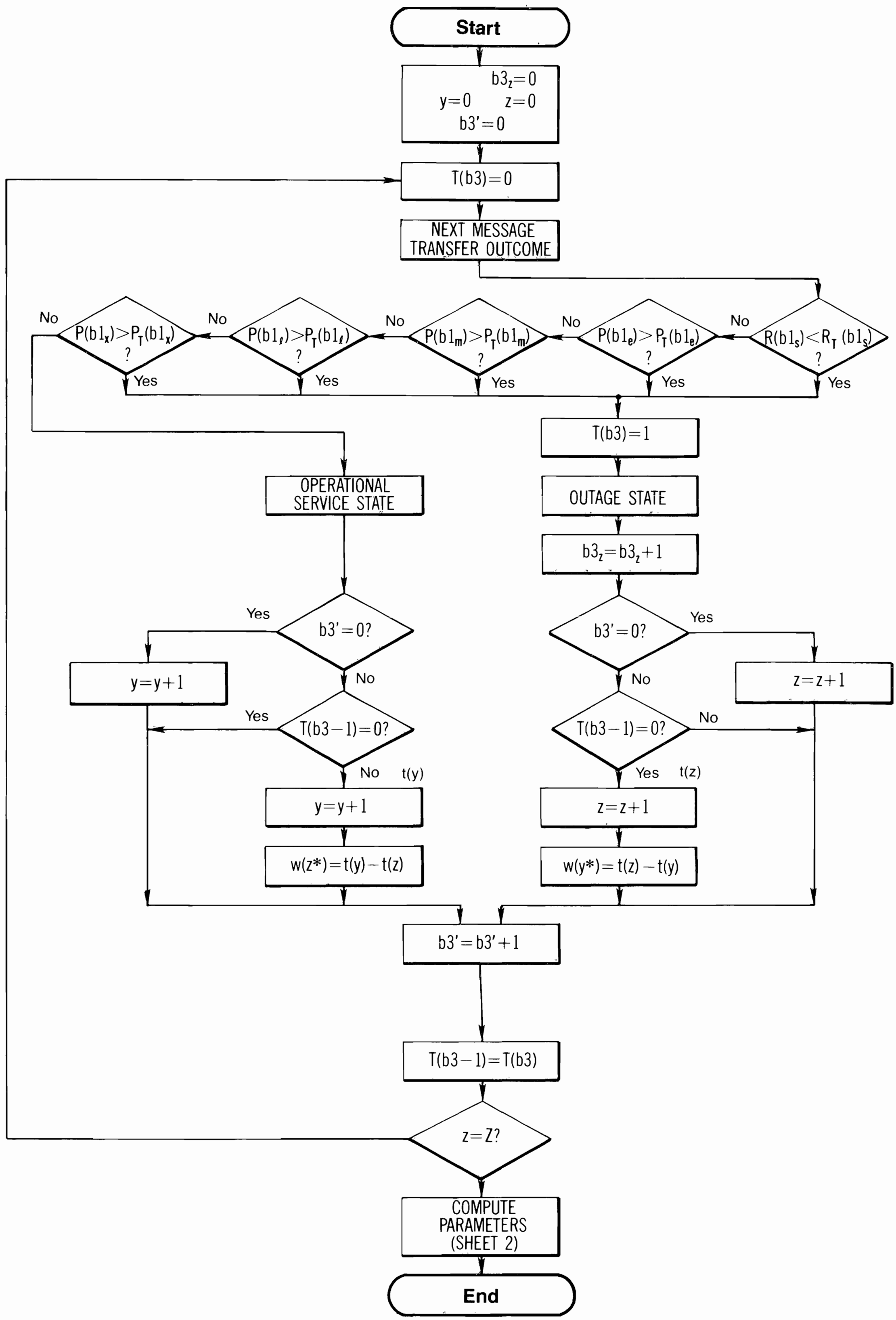


Figure 4.4 (sheet 1). Secondary parameter definition flowchart.

SECONDARY PARAMETERS

1. Service Time Between Outages $= W(y^*) = \frac{1}{Y} \sum_{y=1}^{Y} w(y^*)$

2. Outage Duration $= W(z^*) = \frac{1}{Z} \sum_{z=1}^{Z} w(z^*)$

3. Outage Probability $= P(Z) = B3_z / B3'$

DEFINITION OF CONSTANTS

$B3'$ = Total number of message transfer trials counted during a secondary parameter measurement.

Z = Total number of outage periods counted during a secondary parameter measurement.

$R_T(bl_s)$ = Threshold value of Bit Transfer Rate.

$P_T(bl_e)$ = Threshold value of Bit Error Probability.

$P_T(bl_m)$ = Threshold value of Bit Misdelivery Probability.

$P_T(bl_\ell)$ = Threshold value of Bit Loss Probability.

$P_T(bl_x)$ = Threshold value of Extra Bit Probability.

DEFINITION OF VARIABLES

$b3'$ = Message transfer trial counter. The count excludes trials which fail due to user non-performance.

$T(b3)$ = Aggregate outage state variable for the current performance measurement period (b3).

$T(b3-1)$ = Aggregate outage state variable for the previous performance measurement period (b3-1).

$R(bl_s)$ = Measured Bit Transfer Rate value.

$P(bl_e)$ = Measured Bit Error Probability value.

$P(bl_m)$ = Measured Bit Misdelivery Probability value.

$P(bl_\ell)$ = Measured Bit Loss Probability value.

$P(bl_x)$ = Measured Extra Bit Probability value.

y = Service continuation function (Operational Service period) counter.

Y = Total number of Operational Service periods counted during a secondary parameter measurement.

z = Service Restoral Function (Outage period) counter.

$b3_z$ = Outage state counter.

$B3_z$ = Total number of Outage state outcomes counted during a secondary parameter measurement.

$t(y)$ = Time most recent Operational Service transition occurred.

$t(z)$ = Time most recent Outage transition occurred.

$w(y^*)$ = Value of Service Time between Outages measured on a particular transition to the Outage state.

$w(z^*)$ = Value of Outage Duration measured on a particular transition to the operational Service state.

Figure 4.4 (sheet 2). Secondary parameter definition flowchart.

In either case, the program proceeds to determine whether the new reliability state is different from the previous state, T(b3-1). If a state transition has occurred, the program increments the appropriate transition counter (y or z); and calculates a waiting time value, w(y*) or w(z*), for time spent in the old state. The program then increments the message transfer function counter, b3'; stores the calculated value of T(b3) in T(b3-1) for later comparison; and returns to process the next message transfer outcome. When the number of outage state transitions z reaches a predetermined value Z, the program proceeds to calculate values for the three secondary parameters, based on the equation definitions presented in sheet 2 of Figure 4.4.

5. ANCILLARY PARAMETERS

5.1. Introduction

Section 3 pointed out that the selected primary functions are (in general) user dependent; and made reference to four Responsibility Indicators, which were used to ascribe responsibility for non-performance of individual primary functions to the telecommunication system or the users. This section has two objectives:

1. To define a set of four "ancillary" performance parameters, which describe the influence of user delay on total performance time for each of four associated primary functions[8].
2. To define the four Responsibility Indicators, on the basis of specified (average) and observed (individual) ancillary parameter values.

---

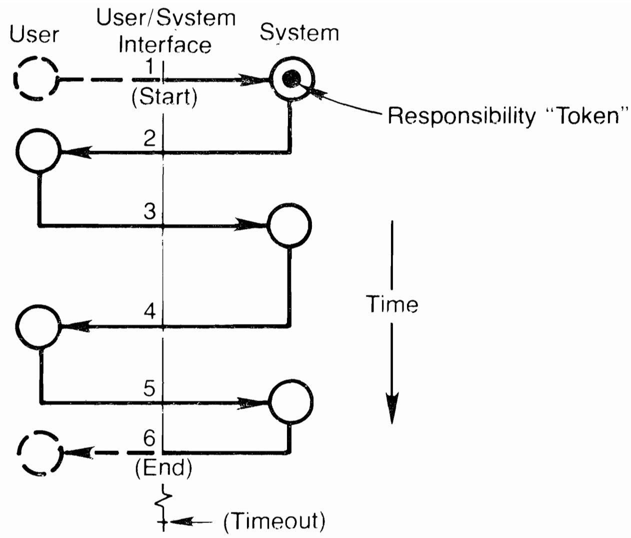
[8]No ancillary parameter is defined for the bit transfer function, since the Lost Bit and Refused Bit outcomes are distinguished by reference to the Responsibility Indicator for the associated block.

The section is divided into two major subsections. The first describes a general sample space of "responsibility outcomes" which was used in defining the ancillary parameters. The second defines the four selected ancillary parameters and their corresponding Responsibility Indicators.

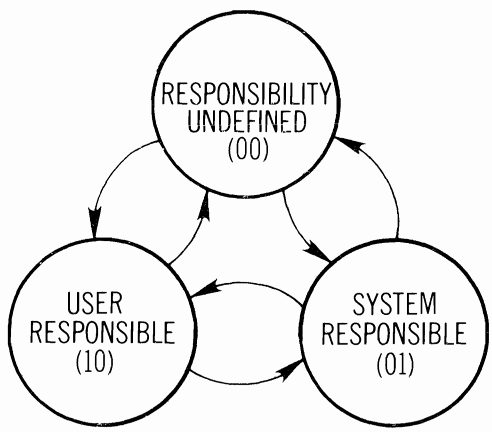
5.2. Ancillary Outcomes

The telecommunication process has been modelled as a sequence of discrete interface events, each produced by unilateral activity on the part of an individual user or half-system entity. At any point in a sequence of interface communications between adjacent entities, it is therefore possible to say that one of the two entities is uniquely "responsible" for producing the "next" interface event. This responsibility concept has been implemented in a pair of complementary ancillary sub-states within the Committed and Closing Transaction states: the Active state, associated with the responsible entity; and the Waiting state, associated with the adjacent (non-responsible) entity. This subsection uses these concepts to define an ancillary sample space which represents all possible distributions of primary function performance time between participating user and half-system entities.

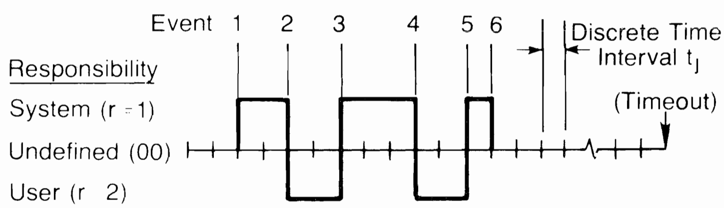
In order to describe the ancillary sample space as clearly as possible we begin with the simplest case, in which the primary function of interest (g) involves interactions at only one of the two user interfaces. Figure 5.1a shows a typical sequence of responsibility transfers which might occur during the performance of such a function. In this graph, the circles represent activities (entity states); and the arrows represent responsibility transfer events. The function starts with Event 1 and ends with Event 6. The sequence of responsibility transitions can be visualized as the movement of a responsibility "token" down the indicated chain of activities.



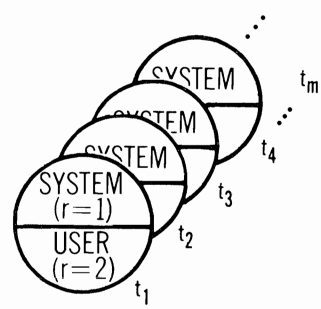
a. Typical Responsibility Transfer Sequence



b. State Diagram



c. Timing Diagram



d. Sample Space

Figure 5.1. Single interface function and ancillary sample space.

From the standpoint of responsibility for the function illustrated, three distinct time periods exist:

1. The period prior to Event 1. Responsibility for completing the function is undefined during this period, since the objective of completing the function cannot exist until the function has been started.
2. The period between Events 1 and 6. Responsibility alternates between the two communicating entities as the token moves down the event chain. Responsibility for producing the next interface event always rests with one entity or the other.
3. The period after Event 6 (or after expiration of the maximum performance time for the function, if Event 6 does not occur within this time). Responsibility for completing the function is again undefined.

Relative to the function being considered, then, there are three possible responsibility states (r) which can be jointly assumed by the two communicating entities:

User State	Half-System State	Joint State (r)	Meaning
1	0	2	User Active; Half-system Waiting
0	1	1	User Waiting; Half-system Active
0	0	0	Responsibility Undefined

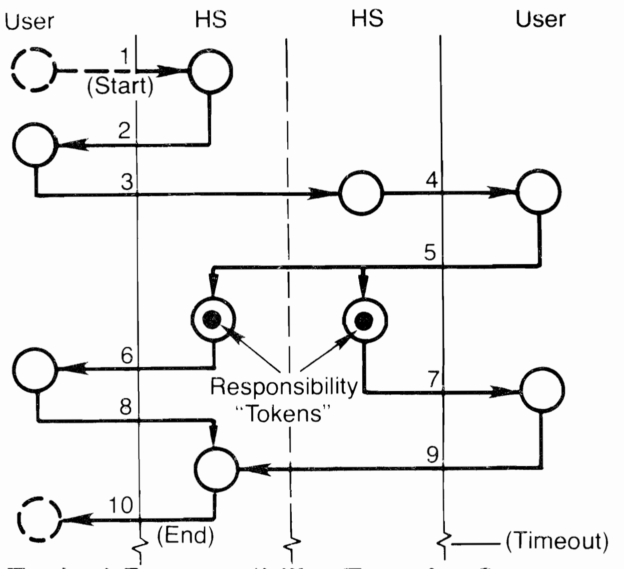
These alternative states and the possible transitions between them are illustrated in Figure 5.1b.

Figure 5.1c represents the postulated responsibility state sequence in timing diagram form; and provides a basis for defining an ancillary sample space for any single-interface primary function (Fig. 5.1d). We divide the function performance period into a succession of discrete time intervals, $\{t_j | 1 \le j \le m\}$; and view each time interval as a "trial" having one of two discrete outcomes: the joint responsibility states r=1 and r=2, as defined above. The outcome r=0 is excluded from the ancillary sample space representing the function since it can only occur outside the function performance period. The ancillary sample space for the single-interface primary function is then defined as a product space, representing the composite experiment of observing the joint responsibility state during each time interval within the performance period. The individual sample points in this space can be viewed as binary vectors, $\{r_1 r_2 r_3 \ldots r_m\}$, each element r_j denoting the responsibility state during an individual time interval j. This sample space represents all possible distributions of performance time between the user and system entities participating in a single-interface primary function.

We now consider the more general case, in which the primary function of interest involves interactions at both user/system interfaces. Figure 5.2a shows a typical sequence of responsibility transfers which might occur during the performance of this type of function. The graph differs from the single-interface graph of Figure 5.1a in two respects:

1. The "next event" in the function event sequence can be associated (in general) with either of the two user interfaces. It is thus possible for both entities attached to a particular interface to be in the Waiting state during the performance of a function - waiting for an event at the <u>other</u> interface. A familiar example is the case, in the public switched network, where both the calling party and the telecommunication system are waiting for the called party to answer.
2. Activities can go on concurrently at the two user interfaces.

A consequence of these differences is that the responsibility state transitions at the two user interfaces are dependent in some situations (e.g., Event 3 in Fig. 5.2a) and independent in others (e.g., Events 6 and 7). In the first case, only a single responsibility token is present in the function graph; in the second case, two tokens are present. Tokens "split" on occurrence of any event that initiates activity at both user interfaces



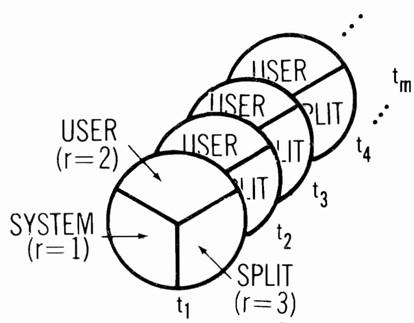
a. Typical Responsibility Transfer Sequence

UHHU \ UHHU		USER 0001	USER 1000	USER 1001	SYSTEM 0010	SYSTEM 0100	SYSTEM 0110	SPLIT 0101	SPLIT 1010	UNDEFINED 0000
USER	0001	–			√	√	√	√		√
	1000		–		√	√	√		√	√
	1001	√	√	–				√	√	
SYSTEM	0010	√			–	*	*	√		√
	0100		√		*	–	*		√	√
	0110				*	*	–	√	√	
SPLIT	0101	√		√		√	√	–		
	1010		√	√	√		√		–	
UNDEFINED	0000	√	√		√	√				–

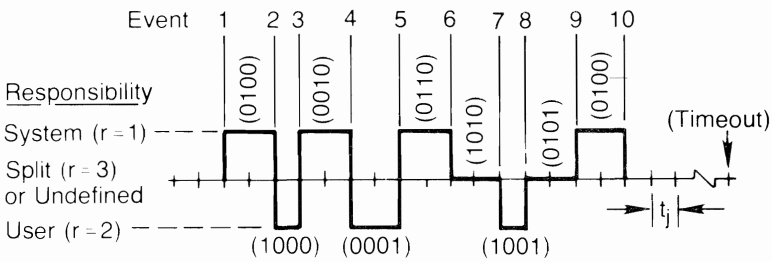
*System-Internal Transition

√ Valid Transition

b. State Transition Table



d. Sample Space



c. Timing Diagram

Figure 5.2. Two-interface function and ancillary sample space.

(e.g., Event 5); and "merge" on occurrence of any event that terminates activity at one user interface (e.g., Event 9). In the example illustrated, activity is underway only at interface 1 between Events 1 and 3; activity is underway only at interface 2 between Events 3 and 5; activities are underway at both interfaces between Events 5 and 9; and activity is again underway only at interface 1 between Events 9 and 10.

The message transfer function provides a practical example of concurrent, independent communication activities at two user interfaces relative to a single function. A "message" typically comprises many separate blocks; at the same time that one block is being transmitted at the source interface, a previously transmitted block may be in the process of being received at the destination.

Consider now the possible joint responsibility states of all four model entities. There are nine valid combinations:

	Entity Responsibility States				Aggregate
#	User	HS	HS	User	Responsibility State
1	0	0	0	1	User
2	1	0	0	0	User
3	1	0	0	1	User
4	0	0	1	0	System
5	0	1	0	0	System
6	0	1	1	0	System
7	0	1	0	1	Split
8	1	0	1	0	Split
9	0	0	0	0	Undefined

In the first three joint states, aggregate responsibility for advancing the function to completion rests with one or both users. In the fourth through sixth joint states, aggregate responsibility rests with one or both half-systems. In the seventh and eighth joint states, responsibility is "split" between a user and half system; i.e., the user is active at one interface and the system is active at the other. The ninth joint state represents the situation before the function starts and after it ends, when responsibility for that function is undefined. Joint states not listed above are excluded by the premise that only one entity can be active at a given interface at any time.

Figure 5.2b defines the valid transitions between these nine joint states.

Figure 5.2c represents the postulated responsibility state sequence in timing diagram form; and provides a basis for defining an ancillary sample space for any 2-interface primary function (Fig. 5.2d). This sample space is identical to the single-interface space defined above, except that three possible aggregate responsibility states are distinguished: system responsible (r=1); user responsible (r=2); and "split" responsibility (r=3). The individual sample points in this space consist of three-state (ternary) vectors, $\{r_1 r_2 r_3 \ldots r_m\}$, $1 \le r_j \le 3$. This sample space represents all possible distributions of performance time between the user and system entities participating in a 2-interface primary function. It can actually be used to represent the single-interface functions as well, since these are simply a special case in which "split" responsibility cannot occur.

5.3. Ancillary Parameters

This subsection defines the four selected ancillary parameters and their associated Responsibility Indicators. All four ancillary parameters are simple functions of random variables defined on the ancillary sample space described above. We associate three variables with each sample point:

$w(g^*)$ = Elapsed time between the start and the end of an individual primary function performance period.

$w_u(g^*)$ = Total time (within the primary function performance period) during which aggregate responsibility for advancing the function to completion rests with the users; i.e., r=2 in Figure 5.2d.

$w_v(g^*)$ = Total time (within the primary function performance period) during which aggregate responsibility for advancing the function to completion is "split" between a user and a half-system; i.e., r=3 in Figure 5.2d.

The ancillary parameters are calculated on the basis of a population of successful performance trials, G_s. The random variables

$w(g^*)$, $w_u(g^*)$, and $w_v(g^*)$ are thus specialized to their "successful outcome" counterparts, $w(g_s)$, $w_u(g_s)$, and $w_v(g_s)$. We represent the average values of these variables as $W(g_s)$, $W_u(g_s)$, and $W_v(g_s)$, respectively. The ancillary parameters p(g) are defined as ratios of these average values, as follows:

$$p(g) = \frac{W_u(g_s) + 0.5\ W_v(g_s)}{W(g_s)} .$$

The parameters p(g) are termed "user performance time fractions"; they express the average proportion of successful primary function performance time that is attributable to user delay. Periods of unilateral user responsibility contribute to W_u, and are "weighted" at their full value in computing p(g); periods of "split" responsibility contribute to W_v, but are "weighted" at only half their full value to reflect the equal division of responsibility between the user and system entities.

Narrative/symbolic definitions for the four specific ancillary parameters selected for inclusion in the standard are provided below.

<u>User Access Time Fraction p(a)</u> - Ratio of average user access time, $W_u(a_s)+0.5\ W_v(a_s)$, to average Access Time, $W(a_s)$, measured over a sample population of successful access attempts, A_s.

<u>User Block Transfer Time Fraction p(b2)</u> - Ratio of average user block transfer time $W_u(b2_s)+0.5\ W_v(b2_s)$, to average Block Transfer Time, $W(b2_s)$, measured over a sample population of successful block transfer attempts, $B2_s$.

<u>User Message Transfer Time Fraction p(b3)</u> - Ratio of average user message transfer time, $W_u(b3_y)+0.5\ W_v(b3_y)$, to average message transfer time, $W(b3_y)$, measured over a sample population of successful message transfer attempts, $B3_y$.[9]

[9] "Successful" message transfer attempts are those encountering the secondary outcome Operational Service state ($b3_y$).

User Disengagement Time Fraction p(d) - Ratio of average user disengagement time, $W_u(d_s)+0.5\ W_v(d_s)$, to average Disengagement Time, $W(d_s)$, measured over a sample population of successful disengagement attempts, D_s.

The purpose of the ancillary parameters is to describe the influence of user delay on total primary function performance time. There are two possible results of user delay on the performance of a primary function:

1. The user delay may simply extend the function performance time, without causing the maximum performance time $3W_N(g_s)$ to be exceeded. In this case, the user delay has its influence on the primary efficiency parameters, which describe successful performance: i.e., the waiting times $W(g_s)$, the time rates $R(g_s)$, and the rate efficiencies $Q(g_s)$.
2. The user delay may combine with the system delay to cause the maximum performance time to be exceeded. In this case, the user delay influences the primary reliability parameters, which describe non-performance: i.e., failure probabilities of the form $P(g_\ell)$.

We address first the influence of user delay on the primary efficiency parameters. Each of the four ancillary parameters can be used directly as a correction factor, to calculate "user-independent" values for the associated efficiency parameters. Given any specified waiting time value $W(g_s)$, the corresponding user-independent value can be calculated as

$$[1-p(g)] \cdot W(g_s).$$

The factor [1-p(g)] is the average system performance time fraction - the complement of p(g). Similarly, given any time rate or rate efficiency parameter, $R(g_s)$ or $Q(g_s)$, the corresponding user-independent value can be calculated as

$$\frac{R(g_s)}{[1-p(g)]} \text{ or } \frac{Q(g_s)}{[1-p(g)]}\ .$$

In each case, the user-independent parameters express the performance that would be provided by the system if user delay were zero; i.e., if all user activities were performed in zero time.

As an example of the calculation of a user-independent parameter value, assume that the User Access Time Fraction for a particular system is specified at 0.6, and that the total Access Time is specified at 25 seconds. The average system access time fraction is then 0.4, and the user-independent value for Access Time is (0.4)(25)=10 seconds. The user-independent rate values are proportionally higher than their user-dependent equivalents because the total performance time required to produce a given number of successful outcomes is shorter in the absence of user delay.

Although the user-independent Efficiency parameters do provide a useful description of system performance, it was concluded that they should not be directly specified in the standard. The basic assumption on which they are defined (zero user performance delay) is not practically attainable; and (as was described in Sec. 3) the user-independent parameters do not reflect differences in the "functional burden" placed on the user by otherwise equivalent services. One familiar example of such a difference is abbreviated vs. normal 7-digit dialing.

We now address the influence of user delay on definition of the primary Reliability parameters. Given any individual performance of a primary function g, it is possible to calculate an individual user performance time fraction,

$$\rho(g) = \frac{w_u(g^*) + 0.5w_v(g^*)}{w(g^*)} \quad ,$$

which represents the outcome of <u>that particular trial</u> from the standpoint of user dependence. The ratio ρ_g is directly analogous to the corresponding ancillary parameter p(g), but differs from it in two respects:

1. The variables in the $\rho(g)$ expression represent individual observed values, rather than averages of a population.
2. The ratio $\rho(g)$ is defined for all primary outcomes, not just for the successful outcomes g_s.

The Responsibility indicator (r_g) value for an individual performance trial is determined by comparing the calculated $\rho(g)$ value for that trial with the corresponding (nominal) ancillary parameter value, $p_N(g)$:

Comparison Outcome	r_g Value	Responsibility
$\rho(g) \leq p_N(g)$	0	System
$\rho(g) > p_N(g)$	1	User

Thus, the variable r_g has the value 0, indicating system responsibility, whenever the user performance time fraction measured on a particular primary function performance trial is less than or equal to the corresponding specified (nominal) value, and conversely. Symbolic definitions for the four specific Responsibility Indicators referenced in the standard are provided below.

Comparison Outcome	r_g Value	Responsibility
$\rho(a) \leq p_N(a)$	$r_a=0$	System
$\rho(a) > p_N(a)$	$r_a=1$	User
$\rho(b2) \leq p_N(b2)$	$r_{b2}=0$	System
$\rho(b2) > p_N(b2)$	$r_{b2}=1$	User
$\rho(b3) \leq p_N(b3)$	$r_{b3}=0$	System
$\rho(b3) > p_N(b3)$	$r_{b3}=1$	User
$\rho(d) \leq p_N(d)$	$r_d=0$	System
$\rho(d) > p_N(d)$	$r_d=1$	User

The purpose of the Responsibility Indicators is to provide a means of ascribing responsibility for non-performance of individual primary functions to the system or the users. Non-performance outcomes are attributed to the system whenever $r_g=0$;

and conversely. Values for r_g are required only when a function is not performed within the specified maximum time, and there is no blocking signal (e.g., UBS or SBS) to identify the entity responsible.

Figure 5.3 is a parameter definition flowchart which provides a procedure for calculating $p(g)$, $\rho(g)$, and r_g values for any primary function, g. Calculation of $p(g)$ is based on a sample population of successful primary function performance trials, G_s. Calculations of $\rho(g)$ and r_g apply to an individual non-performance (g_ℓ or g_f) outcome encountered during performance of a primary function.

To simplify description, we again assume the flowchart is implemented in a computer program. For each distinct responsibility state τ within a primary function performance period, the program (1) determines the current aggregate responsibility state, $r(\tau)$; (2) compares this current state with the previous state, $r(\tau-1)$; and (3) computes the duration of the previous aggregate state, $t(\tau)-t(\tau-1)$, if that state was user responsibility (r=2) or split responsibility (r=3). On the last responsibility transition (to the Function End State, r=0), the program takes one of two paths, depending on the variables being evaluated:

1. If $p(g)$ is being evaluated (for a population of successful trials, G_s) the program increments the successful function counter, g_s; computes values for total performance time $w(g_s)$, total user performance time $w_u(g_s)$, and total "split" performance time $w_v(g_s)$, for the function just examined; and proceeds to the next function. When all G_s primary functions have been examined, the program computes the $p(g)$ value for that sample population, and ends.
2. If $\rho(g)$ and r_g are being evaluated (for an individual non-performance outcome, g_ℓ or g_f) the program can immediately compute these values since only one primary function is being examined. The program then returns

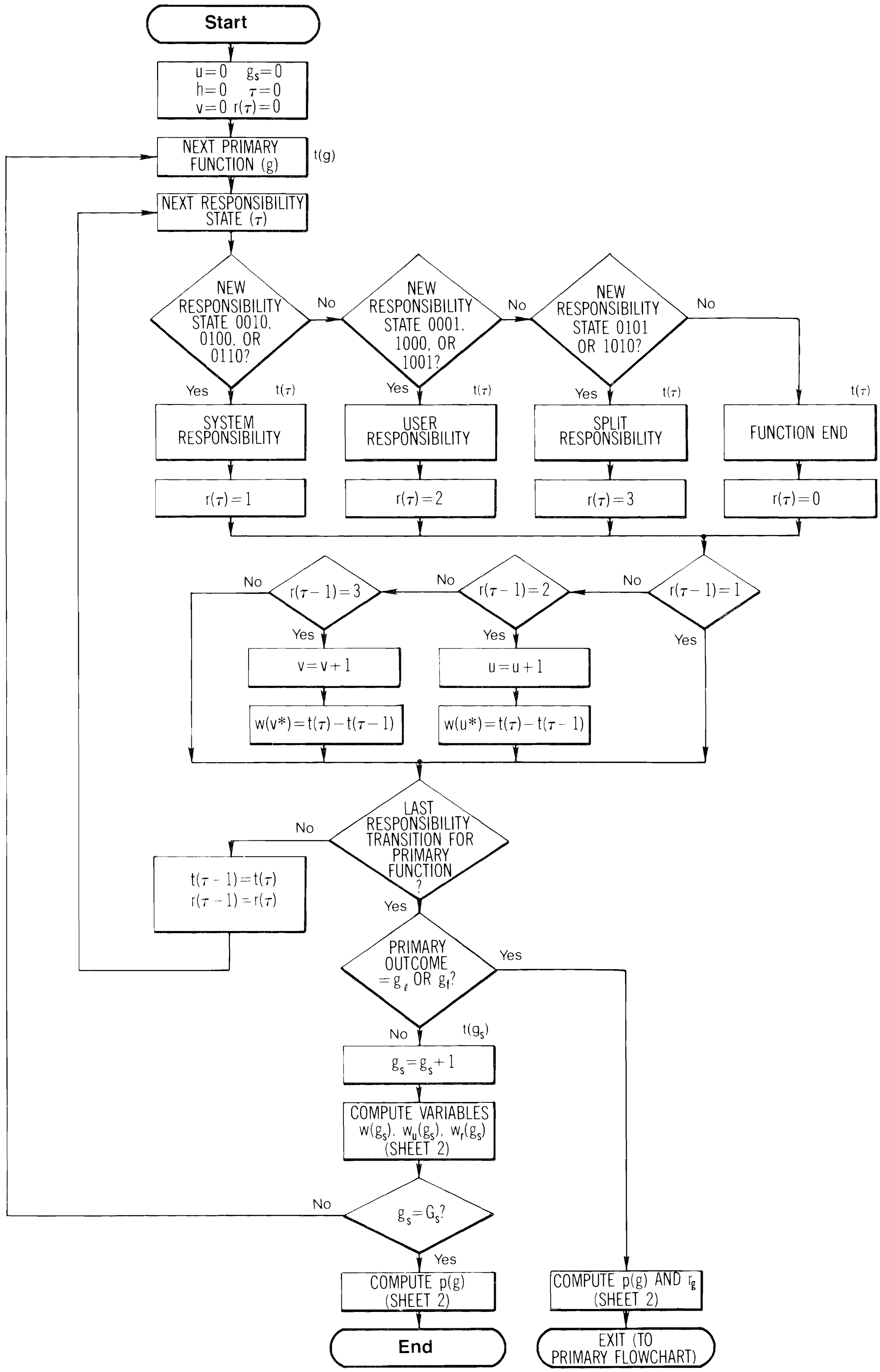


Figure 5.3 (sheet 1). Ancillary parameter definition flowchart

ANCILLARY PARAMETERS p(g)

$$\left\{\begin{array}{l}\text{Average User}\\ \text{Performance}\\ \text{Time Fraction}\end{array}\right\} = p(g) = \frac{W_u(g_s)+0.5\, w_v(g_s)}{W(g_s)}$$

INDIVIDUAL USER FRACTIONS ρ(g)

$$\left\{\begin{array}{l}\text{User}\\ \text{Performance}\\ \text{Time Fraction}\end{array}\right\} = \rho(g) = \frac{w_u(g^*)+0.5\, w_v(g^*)}{w(g^*)}$$

RESPONSIBILITY INDICATORS (r_g)

Comparison Outcome	r_g Value	Responsibility
$\rho(g) \le P_N(g)$	0	System
$\rho(g) > P_N(g)$	1	User

DEFINITION OF CONSTANTS

G_s = Total number of successful primary function performance trials counted during an ancillary parameter measurement.

$P_N(g)$ = Nominal user performance time fraction; the value of p(g) specified by the telecommunication system operator. Describes the proportion of primary function performance time nominally allocated to user activities.

U = Total number of distinct user responsibility states observed during an individual primary function performance period.

$W(g^*)$ = Maximum primary function performance time: $3W_N(g_s)$.

DEFINITION OF VARIABLES

g_s = Successful primary function performance trial counter.

τ = Responsibility state index. Indexes the number distinct responsibility states observed during an individual primary function performance period.

u = User responsibility state index. Counts the number of user responsibility states observed during an individual primary function performance period.

V = Split responsibility state index. Counts the number of split responsibility states observed during an individual primary function performance period.

$r(\tau)$ = Current responsibility state indicator.

$r(\tau-1)$ = Previous responsibility state indicator.

$t(g)$ = Time a particular primary function performance trial starts.

$t(g_s)$ = Time a particular successful primary function performance trial ends

$t(\tau)$ = Time current responsibility state was entered.

$t(\tau-1)$ = Time previous responsibility state was entered ($\tau - 1$).

$w(u^*)$ = Duration of a particular user responsibility state.

$w(v^*)$ = Duration of a particular split responsibility state

$w(g_s)$ = Duration of a particular successful access attempt

$$\left\{\begin{array}{l}W_u(g^*)\\ W_u(g_s)\end{array}\right\} = \sum_{u=1}^{U} w(u^*) \qquad \left\{\begin{array}{l}W_v(g^*)\\ W_v(g_s)\end{array}\right\} = \sum_{v=1}^{V} w(v^*)$$

$$W_u(g_s) = \frac{1}{G_s}\sum_{g_s=1}^{G_s} w_u(g_s)$$

$$W_v(g_s) = \frac{1}{G_s}\sum_{g_s=1}^{G_s} w_v(g_s)$$

$$W(g_s) = \frac{1}{G_s}\sum_{g_s=1}^{G_s} w(g_s)$$

NOTES

1. The flowchart defines all quantities in terms of a general primary function, g. To define corresponding quantities for specific functions, substitute the appropriate function index (a, b2, b3, or d) for g in each instance.

2. The flowchart can be used either to calculate p(g) values, for a sample population of successful performance trials, G_s; or to calculate ρ(g) and r_g values for individual non-performance outcomes (g_v or g_f).

et 2). Ancillary Parameter Defin

Figure 5.3 (sheet 2). Ancillary parameter definition flowchart.

the calculated r_g value to the primary function flowchart from which it was entered, and ends.

Sheet 2 of Figure 5.3 presents equation definitions for the selected ancillary parameters and Responsibility Indicators.

Figure 5.4 summarizes all performance parameters selected for inclusion in the proposed Federal Standard. A total of 26 parameters are specified, including 19 primary parameters; 3 secondary parameters; and 4 ancillary parameters.

In closing, we note one user dependence issue deferred to future study: the influence of user performance on measured values for the accuracy parameters. The foregoing has been based on the assumption that the users and the performance assessment entity are one and the same; and this is normally the case, since performance assessment is a data processing function. But the users, like any measurement instrument, can make measurement errors; and such errors should not be "charged" to system performance. A thorough treatment of this problem would require that user accuracy be evaluated, in order to place bounds on the ability of users to assess system performance.

6. ACKNOWLEDGEMENTS

Proposed Federal Standard 1033 is the result of a cooperative effort involving significant contributions from many members of FTSC and Technical Subcommittee 2. We acknowledge specifically the prior contributions of Marshall Cain and Dennis Bodson of the National Communications System; Dana Grubb of the National Bureau of Standards; Paul Emmons of the National Aeronautics and Space Administration; Howard Berkowitz of the Library of Congress; and Ed Stephan of the General Services Administration. The work has also benefitted substantially from close cooperation with two related standards efforts: the work of American National Standards Institute (Task Group X3S3.5) in developing performance standards for systems using bit-oriented communication control procedures; and the FCC's inquiry into

FUNCTION	PERFORMANCE CRITERION: EFFICIENCY	PERFORMANCE CRITERION: ACCURACY	PERFORMANCE CRITERION: RELIABILITY	
ACCESS (a)	• ACCESS TIME $W(a_s)$	• INCORRECT ACCESS PROBABILITY $P(a_m)$	• ACCESS DENIAL PROBABILITY $P(a_\ell)$	• USER ACCESS TIME FRACTION p(a)
BIT TRANSFER (b1)	• BIT TRANSFER TIME $W(b1_s)$	• BIT MISDELIVERY PROBABILITY $P(b1_m)$ • BIT ERROR PROBABILITY $P(b1_e)$ • EXTRA BIT PROBABILITY $P(b1_x)$	• BIT LOSS PROBABILITY $P(b1_\ell)$	• USER BLOCK TRANSFER TIME FRACTION p(b2)
BLOCK TRANSFER (b2)	• BLOCK TRANSFER TIME $W(b2_s)$	• BLOCK MISDELIVERY PROBABILITY $P(b2_m)$ • BLOCK ERROR PROBABILITY $P(b2_e)$ • EXTRA BLOCK PROBABILITY $P(b2_x)$	• BLOCK LOSS PROBABILITY $P(b2_\ell)$	
MESSAGE TRANSFER (b3)	• BIT TRANSFER RATE $R(b1_s)$ • BIT RATE EFFICIENCY $Q(b1_s)$ • BLOCK TRANSFER RATE $R(b2_s)$ • BLOCK RATE EFFICIENCY $Q(b2_s)$	• OUTAGE PROBABILITY $P(b3_\ell)$		• USER MESSAGE TRANSFER TIME FRACTION p(b3)
DISENGAGEMENT (d)	• DISENGAGEMENT TIME $W(d_s)$	• DISENGAGEMENT DENIAL PROBABILITY $P(d_\ell)$		• USER DISENGAGEMENT TIME FRACTION p(d)
SERVICE CONTINUATION (y)	• SERVICE TIME BETWEEN OUTAGES $W(y^*)$			
SERVICE RESTORAL (z)	• OUTAGE DURATION $W(z^*)$			

Legend
- Primary Parameters
- Secondary Parameters
- Ancillary Parameters

Figu..e 5.4. Summary of selected performance parameters.

the quality and reliability of specialized communication services (Docket 18920, issue D).

We acknowledge the contribution of W.A. Kissick to the telecommunication process model of Section 2; and thank Patricia Sanchez for her assistance in typing and manuscript preparation.

7. REFERENCES

AGREE (1957), Reliability of military electronic equipment, Advisory Group on the Reliability of Electronic Equipment, U.S. Government Printing Office, Washington, D.C.

ANSI (1974), Determination of performance of data communication systems, ANSI X3.44-1974.

ANSI (1975), Proposed American national standard for advanced data communication control procedures (ADCCP), ANSI X3S3.4/589, 15 August.

ANSI (1976), Proposed American National Standard X3S3.7-75-54/3 ("ANSI X.21"), General purpose interface between data terminal equipment and data circuit terminating equipment for synchronous operation on public data networks, June.

AT&T (1971), Bell System Technical Reference Pub. 41007, 1969-70 switched telecommunications network connection survey, April.

AT&T (1975), Comments in response to the FCC's Further Notice of Inquiry and Proposed Rule Making on the Specialized Common Carrier Inquiry (Docket 18920), Issue D; submitted May 23.

Bjorner, D. (1970), Finite state automation - definition of data communication line control procedures, Proceedings of the Fall Joint Computer Conference.

CCITT (1973), Data transmission, Green Book, Volume VIII, International Telecommunications Union, Geneva, Switzerland.

CCITT (1977), Recommendation X.25, Interface between data terminal equipment (DTE) and data circuit terminating equipment (DCE) operating in the packet mode on public data networks, Orange Book, Volume VIII.2, International Telecommunications Union, Geneva, Switzerland.

Cole, G.D. (1971), Performance measurements on the ARPA network, Proceedings of the Second Symposium on Problems in the Optimization of Data Communication Systems, October.

Computer News (1977), NCR details distributed network plans, 22:1143, August 1.

Crow, E.L. (1974), Confidence limits for digital error rates, Office of Telecommunications Report No. 74-51, U.S. Dept. of Commerce, Boulder, CO 80303, November.

Crow, E.L. (1975), Confidence limits for small probabilities from Bernoulli trails, Communications in Statistics, 4(5).

Crow, E.L., and M.J. Miles (1976), A minimum cost, accurate statistical method to measure bit error rates, Proceedings of the International Conference on Computer Communications, Toronto, Canada, August.

Crow, E.L., and M.J. Miles (1977), Confidence limits for digital error rates from dependent transmissions, Office of Telecommunications Report No. 77-118, U.S. Dept. of Commerce, Boulder, CO 80303, March.

Datamation (1976), Problems and predictions for communications processing, June.

Duffy, F.P., and R.A. Mercer (1978), A study of network performance and customer behavior during direct-distance-dialing call attempts in the U.S.A., Bell System Technical Journal, Vol. 57, January.

FCC (1976), Proposed Amendments, Docket 20828 (Second Computer Inquiry), FCC 76-745, released August 9.

FitzGerald, J. (1977), Audit and control of data communications networks, a concensus report, Proceedings of the NBS Invitational Workshop on Audit and Evaluation of Computer Security, Part X, NBS Special Publication 500-19, October.

Goldberg, S. (1960), Probability: An Introduction (Prentice-Hall, Inc., Englewood Cliffs, NJ).

Gray, J.P. (1972), Line control procedures, Proceedings of the IEEE, Vol. 60, November.

IBM (1976), Systems network architecture format and protocol reference manual: Architecture logic, SC30-3112-0, File No. 5370-09, March.

Kleinrock, L. (1976), Queueing systems, volume II, Computer applications (John Wiley & Sons, Inc., New York, N.Y.).

McFadyen, J.H. (1976), Systems network architecture: an overview, IBM Systems Journal, Vol. 15, No. 1.

McManamon, P.M., R.K. Rosich, J.A. Payne, and M.J. Miles (1975), Performance criteria for digital data networks, Office of Telecommunications Report No. 75-54, U.S. Dept. of Commerce, Boulder, CO 80303, January.

Minsky, M.L. (1967), Computation: Finite and infinite machines, (Prentice-Hall, Inc., Englewood Cliffs, NJ).

National Communications System (1977), Proposed Federal Standard 1033, Telecommunication: digital communication performance parameters, Federal Register, Vol. 42, No. 35, February 22.

Payne, J.A. (1978), ARPANET host to host access and disengagement measurements, National Telecommunications and Information Administration Report No. 78-3, U.S. Dept. of Commerce, Boulder, CO 80303.

Schwager, A.O. (1978), The Hewlett-Packard distributed system network, Hewlett-Packard Journal, March.

Seitz, N.B., and P.M. McManamon (1976a), Review of responses to the FCC quality and reliability inquiry (Docket 18920), Office of Telecommunications Report No. 76-101, U.S. Dept. of Commerce, Boulder, CO 80303, August.

Seitz, N.B., and P.M. McManamon (1976b), Performance measures for digital communication services, Business Communication Review, March-April.

Seitz, N.B., and P.M. McManamon (1977a), A systematic approach to the development of performance parameters, Proceedings of the International Telecommunication Exposition (INTELCOMM 77), October.

Seitz, N.B., and P.M. McManamon (1977b), Digital communication performance parameters, Proceedings of the International Telecommunication Exposition (INTELCOM 77), October.

Shannon, C.E. (1948), A mathematical theory of communication, Bell System Technical Journal, Vol. 27, July.

Stone, H.S. (1973), Discrete mathematical structures and their applications, Science Research Associates, Inc., Chicago, IL.

Stutzman, B.W. (1972), Data communication control procedures, Computing Surveys, Vol. 4, December.

FORM OT-29 (3-73)

U.S. DEPARTMENT OF COMMERCE
OFFICE OF TELECOMMUNICATIONS

BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION OR REPORT NO.	2. Gov't Accession No.	3. Recipient's Accession No.
	NTIA-Report-78-4		

4. TITLE AND SUBTITLE DIGITAL COMMUNICATION PERFORMANCE PARAMETERS FOR PROPOSED FEDERAL STANDARD 1033 VOLUME I: STANDARD PARAMETERS	5. Publication Date May 1978 6. Performing Organization Code NTIA/ITS
7. AUTHOR(S) Neal B. Seitz and Peter M. McManamon	9. Project/Task/Work Unit No.
8. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Department of Commerce National Telecommunications & Information Administration Institute for Telecommunication Sciences 325 Broadway, Boulder, CO 80303	10. Contract/Grant No.
11. Sponsoring Organization Name and Address U.S. Department of Commerce National Telecommunications & Information Administration 1325 G Street, NW Washington, D.C. 20005	12. Type of Report and Period Covered 13.

14. SUPPLEMENTARY NOTES

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

There is a growing need within the Federal Government for uniform means of specifying and measuring the performance of data communication systems from the point of view of the digital services delivered to the end user. This report presents results of a cooperative interagency program aimed at meeting this need through the development of universally applicable, user-oriented performance parameters. The defined parameters have been incorporated in a proposed Federal Standard which is currently undergoing review and coordination within government and industry.

16. Key Words *(Alphabetical order, separated by semicolons)*

17. AVAILABILITY STATEMENT	18. Security Class *(This report)*	20. Number of pages
[X] UNLIMITED.	Unclassified	175
[ ] FOR OFFICIAL DISTRIBUTION.	19. Security Class *(This page)* Unclassified	21. Price

USCOMM-DC 29716-P73

☆ U.S. GOVERNMENT PRINTING OFFICE:1978—778-062 / 357

FORM OT-29
(3-73)

U.S. DEPARTMENT OF COMMERCE
OFFICE OF TELECOMMUNICATIONS

BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION OR REPORT NO.	2. Gov't Accession No.	3. Recipient's Accession No.
	NTIA-Report-78-4		

4. TITLE AND SUBTITLE
Digital Communication Performance Parameters for Proposed Federal Standard 1033 - Volume II: Application Examples

5. Publication Date
May 1978

6. Performing Organization Code
NTIA/ITS

7. AUTHOR(S)
F. George Kimmett and Neal B. Seitz

9. Project/Task/Work Unit No.

8. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. Department of Commerce
National Telecommunications and Information Administration
Institute for Telecommunication Sciences
325 Broadway, Boulder, CO 80303

10. Contract/Grant No.

11. Sponsoring Organization Name and Address
U.S. Department of Commerce
National Telecommunications and Information Administration
1325 G Street, NW
Washington, D.C. 20005

12. Type of Report and Period Covered

13.

14. SUPPLEMENTARY NOTES

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

This volume presents three examples of the use of proposed Federal Standard 1033, "Digital Communication Performance Parameters, in specifying end-to-end telecommunication system performance. The examples illustrate, for the particular configurations assumed, the processes of defining user/system boundaries and interaction sequences; identifying communication functions and their associated outcomes; and calculating performance parameter values.

16. Key Words *(Alphabetical order, separated by semicolons)*

access; data communications; disengagement; federal standard; information transfer; performance parameters; user information transfer; circuit switching; message switching

17. AVAILABILITY STATEMENT

[X] UNLIMITED.

[ ] FOR OFFICIAL DISTRIBUTION.

18. Security Class *(This report)*	20. Number of pages
Unclassified	66
19. Security Class *(This page)*	**21. Price:**
Unclassified	