Applications of Processing State Transition Diagrams to Traffic Engineering

Peter Gerrand
Telecom Australia*, Melbourne, Australia

ABSTRACT
The likely adoption by the Vth Plenary Assembly in October 1976 of the graphical Specification and Description Language (SDL) prepared by the CCITT's Study Group XI offers potential advantages to teletraffic engineers. This paper introduces the SDL, and suggests a general scheme for the systematic application of the SDL to system documentation, whereby the documentation required for capacity studies can be generated as a natural part of the system design process. The usefulness of processing state transition diagrams in general, of which the SDL is a special but important case, to both simulation and analysis of traffic capacity is discussed.

1. INTRODUCTION
Telephone switching systems have always been complicated things, and their increasing versatility with stored program control (SPC) has not made their behaviour any the easier to understand. One can highlight three stages in the life of an SPC telephone exchange in which deep understanding of its working is required. During the stages of acceptance testing weaknesses in the manufacturer's design or documentation, and weaknesses in the administration's specification, are usually discovered and corrected. During and after major extension of the exchange to incorporate new facilities or attain a larger capacity, the system documentation may be found to be incomplete or inaccurate. Thirdly, whether carried out during the initial phase of system development or at any later period in the life of the exchange, a traffic capacity study using a simulation model or other means will only be accurate if the traffic engineers have absorbed a deep and detailed understanding of the system. In all three stages, the accuracy, completeness and intelligibility of the system documentation are obviously of critical importance.

In February 1976, the CCITT's Study Group XI finalized a draft Recommendation on the use of a graphical Specification and Description Language (SDL) for application to the documentation of SPC switching systems. By the time this International Teletraffic Congress commences, it will be known whether the Plenary Assembly of the CCITT has approved this Recommendation or not. All the auguries point to its approval, in which case a new international standard will have been created, and one can expect that an increasing number of SPC switching systems will be documented using this SDL. Section 3 of this paper will provide an introduction to the SDL, and Sections 4 and 5 will indicate its potential usefulness to traffic engineers in the simulation, analysis and general understanding of switching systems - both SPC and non-SPC.

But first it will be necessary to devote a short section, Section 2, to a simple classification scheme for levels of system documentation, which will make it easier to explain the various applications of the SDL, and in particular to make clear what is meant by "specification" and "description".

2. LEVELS AND PARTITIONS IN SYSTEM DOCUMENTATION
In the context of the purchase of telephone switching systems, a clear distinction exists between a specification and a description of the behaviour of a system. A functional specification shows the requirements of a system in terms of its behaviour in response to several given sequences of inputs. A functional description shows the actual behaviour of a realized system in response to all possible sequences of inputs, giving considerable detail concerning the internal structure of the system.

Thus a specification normally precedes design, and views the behaviour of the system "from the outside", whereas a description normally follows design, and describes the behaviour of the system "from the inside".

However, in the context of the development of the system, the design process may be observed to pass through several phases, in which the same document may serve both as a description (of the design decisions made up to this phase) and as a specification (for further, more detailed design).

Amongst the many levels of documentation which characterise different phases of switching systems design, three levels appear to be fundamental, irrespective of whether the documentation language is of narrative or diagrammatic form.

A Level 1 document is a specification of the system requirements without prejudice to the choice of switching structure, control structure or technology. Level 1 is generally dependent only upon the interworking requirements between this and other parts of the network. Hence the CCITT R2 and No. 5 signalling specifications are examples of Level 1 documents, largely in narrative form.

A Level 2 document also specifies system requirements, but it is system-dependent to the extent that it introduces the design decisions concerning the structure of the switchblock (e.g. junctors, switching units and signalling units) and the peripheral units (e.g. markers, scanners and drivers) that will drive the switchblock. Level 2 does not introduce any design decisions concerning the internal structure of the Central Control. Hence Level 2 serves as a description of the system behaviour from the point of view of its switching structure, and as a specification of the behaviour of the Central Control.

Level 3 introduces the design decisions concerning the internal structure of the Central Control (e.g. the internal queuing structure, and the allocation of central memory blocks to call processes, etc.), as well as all design decisions described in Level 2. Thus Level 3 serves as a high-level description of the behaviour of processes in the system; "high-level" by contrast with the flowcharts and lists of coding that serve as the implementation levels in design.

Within any of these Levels, it is convenient to partition the system documentation into several Functional Blocks (using the terminology of the CCITT SDL). Examples of the typical Functional Blocks one might expect to find in each of the three Levels are shown in Figs. 1, 2 and 3. Each Functional Block has only a partial view of what is occurring in the whole system, and each Functional Block must maintain its own consistent point of view: e.g. per-call, per-equipment or per-subsystem. Through the interaction of the Functional Blocks, the behaviour of the total exchange is specified or described.

* This paper was written, and is based upon work performed, while Mr. Gerrand was on leave from Telecom Australia, working as a consultant to the Laboratorios ITT de Electricas, S.A. in Madrid, from November 1974 to October 1976.
Further we note that a full description of the system in Level 3 will normally include Functional Blocks that are never specified in Level 2, but are left to the choice of the designers of the Central Control, e.g. the internal queueing structure of the system. Similarly, the system designers will normally specify certain Functional Blocks in Level 2 that were never specified by the customer in Level 1, e.g. the system recovery and initialization tasks.

The structure of Levels and Functional Blocks which has been introduced in this Section is independent of the language chosen for specification and description; it serves as a framework in which the techniques of a graphical language such as the CCITT SDL can be flexibly employed.

3. INTRODUCTION TO THE SPECIFICATION AND DESCRIPTION LANGUAGE

3.1 THE CONCEPT OF A PROCESS

The central concept in the CCITT SDL is that of a process. This word is not formally defined; its meaning is entirely compatible with its use in ordinary English, and also with its use in the theory of stochastic processes. Some relevant examples of processes are: any kind of telephone call process; a signal-recognition process; a system recovery process. In short, the names of the different Functional Blocks shown in Figs. 1, 2 and 3 indicate the nature of the processes belonging to these Blocks. A Functional Block may contain one or more processes, but to avoid confusion we insist that any given process may only belong to one Functional Block, in any given Level.

It is typical of a process that it passes through several states. Often it may be considered as a life-and-death process, in which its birth occurs when it leaves some "idle" or "initial" state, and its death occurs when it returns to that original state or goes to some permanent terminal state; such a process may have any number of lives. For example, a call-process may be created by assigning a word of memory to an incoming line, initialized with a memory state meaning "line equipped, in service, and idle". Each time the memory state changes from "idle" to "seizure", a new call is said to be born and only when it returns to the idle state is that call considered to have died.

3.2 BASIC DEFINITIONS IN THE CCITT SDL (Ref. 1)

**Signals**: A signal is a flow of data conveying information to or from a process. A signal may be either in hardware or in software form. If the information flow is from a process described by a Block to a process described by another Block it is an external signal. If the flow is between processes described by the same Block it is an internal signal.
Outputs: an output is an action within a transition which generates a signal which in turn acts as an input elsewhere. In accordance with the definition of signals, an output can be either internal or external, but note that the output is the processing action which generates the signal, and not the signal itself.

Decisions: a decision is an action within a transition which asks a question to which the answer can be obtained at that instant and chooses one of several paths to continue the transition.

Tasks: a task is any action within a transition which is neither a decision nor an output.

Fig. 4 Recommended Symbols in the CCITT SDL

3.3 SYMBOLS OF THE CCITT SDL

The basic and recommended symbols of the CCITT SDL are shown in Fig. 4. These graphical symbols are combined to create a diagram that will describe the behaviour of a process. Such a diagram may be called a processing state transition diagram (PSTD): a PSTD is a graph which describes the (required or observed) behaviour of a process in terms of its set of possible states, the processing transitions between these states, and the inputs which cause these transitions. The CCITT SDL is but one of several graphical languages that have been proposed for PSTDs, but it has the merit of being the only language to have achieved international standardization. Particular cases of a PSTD, when applied to a call, an equipment, a queue or to overhead (inter-queue) logic, will be referred to as a CSTD, ESTD, QSTD and OSTD respectively, irrespective of the particular graphical language used to represent the processing transitions.

To be a legitimate PSTD using the CCITT SDL, a necessary condition is that each transition must satisfy the connectivity diagram shown in Fig. 5, starting and ending at a state symbol, passing only once through an input symbol. This condition eliminates illegal combinations such as a dangling transition that never terminates in a state, or transitions without inputs, etc.

The CCITT SDL does not restrict the contents of its major symbols; it specifically includes the option of including state pictures within a state symbol as a means of identifying the exact state of a process and also of simplifying the definition of a sequence of processing actions within a transition. An example of a Level 1 specification:

* The Japanese use the term STD in a sense sufficiently general to include all PSTDs.
tion of a call-processing transition is shown in Fig. 6. The processing required in going from state 4 (ringing) to state 5 (Talking) is the same in Fig. 6a, using state pictures, as in Fig. 6b, not using state pictures. As shown in this example, the total processing involved when passing from one state to the following state is that required to effect the changes in the state pictures, together with the processing indicated in any decisions, outputs or tasks appearing in the transition between the states.

The CCITT has not yet recommended the use of particular symbols for the pictorial elements appearing within state pictures: this is the subject of further study by the CCITT's Study Group XI in 1977-80. This paper uses the symbols for pictorial elements that have been preferred in triad use within the Australian administration (Refs. 2-4) for alternative symbols, see Refs. 5-7.

### 4. SYSTEMATIC APPLICATION OF THE SDL TO SYSTEM DOCUMENTATION

The greater usefulness of the SDL to teletraffic engineering is this: that the documentation required for simulation studies and system analysis can be generated as a natural part of the system design process, with equal value to system designers and traffic engineers.

Fig. 7 proposes a scheme for the systematic application of the SDL to system documentation. It assumes a situation in which the customer has no direct involvement with the design of the switching system, and wishes to provide a system independent specification. Functional system specification begins with a set of Level 1 PSTDs, negotiated between customer and supplier. It is expected that all the CCITT international signalling specifications will appear in SDL form in the next few years, and that national signalling specifications will be similarly expressed. To show the power of expression of the SDL, a Level 1 specification for R2-R2 transit signalling will be provided as a handout (Appendix 1) to this paper. For examples of the use of XSTDs in the Level 1 specification of sophisticated local call facilities, see Gale (Ref. 4).

Within the manufacturer's organization, a system specification group can prepare a set of Level 2 PSTDs in which the hardware structure appropriate to a chosen generic design is incorporated. This set of documents needs to be negotiated between the system specifiers and designers before reaching final status as the Level 1 specification for the designers responsible for the Central Control.

The Level 2 CSTD specifications can be produced by a process of systematic modification of the Level 1 CSTDs, requiring relatively little effort. The modification consists of four steps (see Fig. 8 parts (a) and (b)):

1. **Modify the Level 1 state pictures to reflect the choices of switching structure and signalling equipment modules.**
2. **In the Level 1 transitions, insert the appropriate orders for the markers, drivers and other equipment peripheral to the Central Control.**
3. **Relate the signalling states in the Level 1 CSTDs to hardware testpoint conditions in the chosen equipment; then augment the CSTD to include transitions due to those extra testpoint conditions (usually fault conditions) not considered in Level 1.**

* When the customer is directly involved in the design of the switching system, as occurred in the development of the Japanese D10 system, it can be more convenient to use Level 1 specifiers with agreed switching structure (trunking diagram), and hence commence with Level 2 PSTDs. Several examples of such Level 2 specifications are given in Refs. 5-7.

** This step is not strictly necessary if the changes in the state pictures alone are sufficient to determine unambiguously the identities of the peripheral equipment needed to effect these changes as well as the appropriate orders.

### (d) Augment the CSTD to include those equipment-testing routines (such as path continuity tests) that the system designers decide to include as an intrinsic part of a call process.

Once completed, the Level 2 PSTDs serve not only as a specification for the design of the Central Control, but also as a specification for the environmental simulation team, whose job is to test the functioning of the Central Control by simulating its exchange environment. This is made possible because the Level 2 PSTDs include not only the external exchange signals (as in Level 1), but also the output orders to the equipment modules peripheral to the Central Control (see Fig. 6b). In addition, the Level 2 CSTDs can be annotated by traffic engineers to show both call-mix data (the probability that the call will follow any particular transition from a given state) and call-pattern data (the average arrival time of each input signal, plus any additional parameters used to model its arrival time distribution), as valuable input data for the environmental simulation.

The Level 2 PSTDs do not suffice as a system description for capacity studies, since they do not describe the queuing logic that services calls within the system, nor the states in which a call is queued for further processing work, nor the fates of calls when they recognize inputs such as time-outs during queuing states.

This information is found in Level 3 PSTDs, which describes the complete behaviour of processes in the realized system, including all fault conditions except those unpredictable processor faults which would invalidate the description itself. Ideally, from the point of view of traffic engineers, Level 3 PSTDs can be produced by the designers of the Central Control themselves as a natural part of the design process, preceding the design of program flowcharts and coding. Once customers start demanding a full system description using the CCITT SDL, it is likely that the designers will be given the responsibility for the production of the Level 3 diagrams. If the designers are committed to a design process using other documentation techniques, the Level 3 PSTDs can be produced retrospectively as a higher-level description of the logic which the designers have already implemented. This may seem a formidable task, but it can be simplified enormously if the starting point is the Level 2 documentation.

To convert Level 2 PSTDs to Level 3 PSTDs, the following steps can be performed systematically (see Figs. 8b & 8c):

1. **Extend the Level 2 state pictures to include those software cells and timers that are allocated to the process during these states.** (If a software cell is allocated permanently to a process, e.g. a "line
**FIG. 8 COMPARISON OF LEVEL 1, 2 AND 3 DOCUMENTATION OF THE SAME PROCESSING WORK**
equipment category word" for a call, there is no point in adding this information to the state pictures).

(b) Draw and define extra states corresponding to all suspended processing states that may occur in between the Level 2 states. Their state pictures should indicate that further processing action has been scheduled.

(c) For each state in the PSTD, identify the effects of all possible inputs (particularly line signals, register signals, time-outs and further processing actions), showing the processing transitions and the new states reached in every case. The processing transitions should include all decisions that affect the fate of a call, e.g., that decide the extent to which it will be delayed, or whether it will succeed or fail.

In practice, it is useful to annotate the flowlines in the Level 3 PSTDs so as to identify the software programs which are utilized in each transition. In this way, processing actions summarized concisely in the Level 3 PSTDs can be checked against their more detailed descriptions in the lower level system documentation. (These annotations are not shown in Fig. 8c).

5. APPLICATION TO SYSTEM SIMULATION

Perhaps the most general and basic difficulty encountered in the simulation of a switching system occurs when the implementation documentation is not suitable as a functional description for simulation modelling. The traffic engineer is confronted with the flowcharts for perhaps 100 k words of program code; he has to separate the relevant from the irrelevant, partition those programs and routines which will not affect system capacity; he has to re-orient the system description from one concerned with the efficient execution of modularized switching functions concerned with the frequencies of delay and loss of calls through the system. If he does not adequately re-orient the system description, he will become overly preoccupied with measurements that are meaningful to the system designers, but which have only secondary relevance or even no relevance whatsoever, to the calculation of system capacity.

Then, when the traffic engineer has designed and coded a compact simulation model of perhaps 5 k statements, it is likely that his system description - of the simulation model - will be heavily oriented to the particular computer simulation language used, and hence not readily comprehended by the system designers, who will have difficulty in checking the simplifying assumptions introduced into the simulation model, if they are indeed brave enough to attempt to check them.

This may seem a gloomy scenario, but I believe it is more typical than untypical of the state of the art of switching system simulation in the world today, in which the traffic engineer has great difficulty in penetrating the documentation of the real system, and in which the system designers have equal difficulty in penetrating the documentation of the traffic simulation model. This difficulty in communication does not prevent reasonably accurate estimates being obtained for system capacity, but is does lead to the following typical costs and overheads:

(a) changes in the system design usually necessitate changes in the coding of the simulation model, rather than simply changes in the input data to that model;

(b) effectively a new simulation model is designed for each significantly different network application of a switching system.

The thought occurs that, if it is reasonable to assume that the behaviour of any contemporary SPC switching system can be described using a set of PSTDs, then a machine which can read a set of PSTDs and then simulate the behaviour of that set of PSTDs will be a truly generic simulation model, capable of simulating any contemporary SPC switching system. This machine would not only be free from the costs and overheads mentioned in the previous paragraph; it would also have the advantage that the PSTDs can serve as common documentation for the real system and the simulation model itself.

The appropriate documentation would consist of Level 3 PSTDs in which estimates of processing times are given for each significant transition. The significant transitions are those whose probabilities of execution, from the point of view of traffic engineering, are not negligible, and whose execution affects the traffic capacity of the exchange. Exclusion of non-significant transitions from consideration is motivated as much by the desire to minimize the work of the system designers in estimating processing times as to reduce the quantity of input data to the simulation model.

Since the CCITT SDL is currently applicable to sequentially functioning processes only, it would be necessary to first partition the system into its minimum number of sequential machines: thus each central processor, each peripheral device, and even the autonomous system clock may need to be identified separately. Having made this first "horizontal" partition of the system, a second "vertical" partition of the system would be advantageous (see Fig. 9), dividing each sequential machine into three Functional Blocks, whose Level 3 PSTDs are called:

(i) the QSTD (Overhead STD), which describes the inter-queue logic, including for example the scanning logic in a central processor;

(ii) the QSTD (Queue STD), which describes the intra-queue (queue-handling) logic for each queue in this sequential machine;

(iii) The CSTD (Call STD), which describes the call processes as handled by this sequential machine.

The partition of the total queueing structure into an OSTD and QSTD is partially converted into the central processor, in which a multi-queue structure typically occurs. (In the case of the peripheral devices and the system clock, no OSTD is necessary if each of these devices acts as a single-queue or zero-queue sub-system: see Fig. 9). It is believed typical of SPC system design that individual changes in the system design will tend to create changes in only one of the OSTD, QSTD or CSTD but rarely two of these at a time. For example, a change in the overload control strategy is likely to necessitate a change in the OSTD for the central processors alone; and the addition of a new call-handling facility is likely to change the CSTDs alone.

An example of the interaction between parts of the OSTD, QSTD and CSTD for a central processor is shown in Fig. 10.

In order to prepare the diagrams as input data to the simulation, it is of course necessary to obtain estimates of the processing times in all significant transitions in the Level 3 PSTDs. In practice, most of these estimates will be constant times; a small minority will require modelling with a particular random distribution of processing times.

Since the purpose of this Section is to be suggestive rather than prescriptive, it is intended to go into details of the preparation of the Level 3 PSTDs as input data. Sufficient to say that each significant transition in a Level 3 PSTD can be assigned an input data "cell", consisting of a well-defined list of parameters, such as: transition time; output signals to be sent to other processes; the identities of queues or devices whose occupancy counters must be incremented or decremented during this transition; the next state of the process; and the identity of the PSTD from which control is returned. In cases where any of these parameters are not predetermined, special characters can be used to identify library sub-routines which will generate the required data.

During the author's work as a consultant engineer in Madrid in 1975-76, he has contributed to the design of a generic simulation model with a more modest range of application, being content to simulate the behaviour of a particular SPC transit exchange in a wide range of network applications. The scheme of partitions shown in Fig. 9...
was used, and the starting point for the development of the Level 3 PSTDs was the set of Level 2 CSTDs produced by the system specification group. The traffic engineering team found it possible to develop a complete set of relevant CSTDs, PSTDs and OSTD many months in advance of the coding of the real system, by consultation with the system designers. Preliminary estimates of processing times have been inserted in the input data to the simulation model, in advance of the accurate estimates obtainable when the system is fully implemented. In this way, the traffic engineering team is able to produce results which will influence the detailed design decisions, before the system has been implemented as a "fait accompli".

6. APPLICATION TO ANALYSIS

Since the previous International Teletraffic Congress in 1973, several papers have been published, notably Refs. 9 and 10, on the difficult problem of analysing the traffic capacity of an entire local telephone system or telephone network, and its performance during overload conditions. Since these global analyses must take into account customer behaviour, traffic dispersion and all significant forms of congestion in the network, the algebraic forms of their solutions are inevitably very complicated.

It is the author's belief that such global analyses can be significantly aided if a Level 3 CSTD is used as the basic model to define the range of dynamic behaviour of the call processes in the given system. The Level 3 CSTD has the following relevant properties:

(a) it shows every possible fate of the call (excluding those extreme processor fault conditions which would violate the logic of the CSTD) and hence includes the effects of all relevant call-failure mechanisms;
(b) its state pictures enable one to relate the analytical model to the allocation of equipment, supervisory timers and queues to calls in the real system, and hence to manipulate the analytical solution according to changes in the system design;
(c) its properties as a directed graph (or signal-flowgraph) can be exploited to produce a formula for the probability of call-failure, taking into account all significant mechanisms of call-failure.

These ideas have been explored in an earlier paper by the author (Ref. 3); a brief outline will be given here to show their general thrust.

![Fig. 11: Signal-flow graph showing all call set-up paths in a given Level 3 CSTD](Ref. 3, Fig. 1)

A call's progress can be viewed as a random path through a CSTD, starting at the idle state and passing from state to state via the permitted transitions. As an example, Fig. 11 is a subgraph of a Level 3 CSTD (Ref. 3, Fig. 1), reduced to nodes and flowlines, showing the permitted paths from the idle state (node 0) to the conversation state (node 5). Any CSTD using the CCITT SDL can be reduced to such a simple signal-flow graph by replacing the state symbols and decision symbols with numbered nodes, absorbing the other major SDL symbols (inputs, outputs and tasks) into the flowlines that connect them, and discarding all flowline and nodes that are not part of call set-up sequences.

A probability function \( P_m,n \) can be associated with each flowline in a CSTD's signal-flow graph:

**Definition 1:** \( P_m,n \) is the probability that a call currently in node \( m \) will make its next transition to node \( n \).

Using the convenient algebra of signal-flow graphs, the total probability of reaching node 5 from node 0 in Fig. 11 can be written down by inspection:
The system's probability of loss for this type of call might typically be defined as \((1 - P_{0,1})\) under the assumptions that the calling subscriber is "in service" \((P_{1,2} = 1)\) and actually attempts a call \((P_{0,2} = 1)\). The problem remains of relating the individual transition probabilities \(P_{m,n}\) to the basic parameters of the telephone network.

In the cases of signal flow from a decision node, the transition probabilities can generally be evaluated either from traffic dispersal statistics (for such decisions as DIGIT ANALYSIS) or by applying classic traffic theory to link congestion in the exchange (for such decisions as DIGIT RECEIVER AVAILABLE?).

In the case of signal flow from a state node, the transitions are triggered by inputs, which are time-dependent events; the probability of a call taking a particular transition corresponding to event #1 is of course the probability that event #1 occurs before any of the events #2, #3, ... #N.

Definition 2: \(P\{#1 \text{ occurs before any of the events } #2, #3, ... #N\}\) = the probability that an event #1 occurs before any of the events #2, #3, ... #N.

Definition 3: \(q_i(t)\) = the probability that event #1 occurs in the time interval \((0,t)\).

Theorem 1:

For \(N\) statistically independent events #1, #2, ..., #N,

\[ P\{#1 \text{ occurs before any of the events } #2, #3, ... #N\} = \sum_{i=1}^{N} \int_{0}^{\infty} q_i(t) \cdot \prod_{i=1}^{N} (1-q_i(t)) \, dt \]

where the time origin is chosen so that \(q_i(0) = 0\) for all \(i\).

Ref. 3 proves this simple theorem and gives tables of results for (a) pairs of competing events, and (b) trios of competing events, when several arrival distributions, typical for telephony events, are substituted for the \(q_i(t)\).

NOTE: As a consequence of Theorem 1, an additional result, useful for calculating device occupancy, can be given in terms of the already defined probability functions. For a given state with \(N\) statistically independent input events, the time spent in that state can be given a cumulative distribution function \(F(t) = P\{#1 \text{ occurs in the time interval } (0,t)\}\), evaluated as

\[ F(t) = \sum_{i=1}^{N} \int_{0}^{t} q_i(t) \cdot (1-q_i(t)) \, dt \]

Since each state picture indicates the devices occupied during that call state, equation (3) can be used to calculate the distribution function for device occupancy times, if it is valid to assume that the time the call spends in transitions is negligible compared to the time it spends in states.

7. CONCLUDING REMARKS

This paper has introduced a general classification of three levels of system documentation in order to aid discussion of the applications of the CCITT SDL to traffic engineering. (It should be noted that the CCITT's draft Recommendations, Ref. 1 does not itself include any classification scheme for levels of system documentation, although it assumes that the SDL will be applied to more than one level within a hierarchical scheme of system documentation).

Perhaps the key suggestion in this paper is the systematic application of the CCITT SDL to traffic engineering. (It should be noted that the CCITT's draft Recommendations, Ref. 1 does not itself include any classification scheme for levels of system documentation, although it assumes that the SDL will be applied to more than one level within a hierarchical scheme of system documentation).

This proposal is put forward with the hope of benefiting both the customer and the supplier; it is the suggestion of the author as an individual, and has yet to be considered within Telecom Australia.

8. ACKNOWLEDGMENTS

The CCITT SDL, Ref. 1 was prepared by Sub-Group Xi/3-1 of Study Group XI in 1973-76, with the active participation of experts from more than fifteen telecommunications administrations and manufacturers. The author is grateful to Standard Eléctrica, S.A. for releasing him at intervals from his work as a consultant in 1975-76 to attend meetings of Sub-Group Xi/3-1 as a nominated expert of the Australian Administration. In developing ideas for the systematic application of the SDL to system documentation, the author has benefitted from several interchanges of ideas with other members of Sub-Group Xi/3-1. Most of all, as in his previous work (Refs. 2 and 3), the author has been deeply influenced by the multiple possibilities inherent in the use of state pictures, which originated in the early stages of the Japanese DEX projects (Refs. 5 and 6).

For experience gained in applying PSTD's to the specification, description, and simulation of an SPC international transit exchange, the author is grateful to his former colleagues in Standard Eléctrica: Dr. Clementina Bravo for her help in writing the Level 1 CSTD's; Brian Andrews and Luis García-Fernández for their work in writing the Level 2 CSTD's; and Susi Sánchez-Puga, Enrique Martí and Dr. Clementina Bravo for their help in producing the Level 3 PSTD's.

Finally, for fortifying the author's motivation to persevere with the struggle to achieve a massive improvement in the quality of system documentation for the purpose of traffic engineering, he is grateful to Dr. Clem Pratt in Australia, and to Antonio Guerrero and Eduardo Villar in Spain.

9. REFERENCES


1. INTRODUCTION

This Appendix demonstrates the application of the CCITT SDL (Specification and Description Language) to the specification of the call-handling tasks required in R2-R2 transit signalling (with the analogue version of line signalling). The graphical specification of the call-handling tasks, shown in section 4 below, has been extracted from the CCITT narrative Recommendations Q.350-Q.356 and Q.361-Q.368 of the Green Book, Vol. VI, 1972. The effect of pilot failures in the transmission equipment on call-handling has been omitted from this graphical specification, for convenience in preparing this example, but otherwise all sequences of line and register signals that impinge upon call-handling are believed to have been taken into account. The author accepts full responsibility for any errors arising in this example.

That part of the telephone exchange concerned with R2-R2 transit signalling is considered to be partitioned functionally into three Functional Blocks, as follows, without prejudice to methods of implementing the system.

![Functional Blocks Diagram]

2. INPUT AND OUTPUT SIGNALS RELEVANT TO FUNCTIONAL BLOCK 1

2.1 The input signals recognized by the call-handling processes are as follows:

(a) Set \( m_{21} = \{ j_1, j_1, j_0, j_0, d, \bar{d}, b, \bar{b} \} \)

where

- \( j_1 \): incoming junction (line) signal TONE ON;
- \( j_0 \): outgoing junction (line) signal TONE ON;
- \( j_1 \): incoming junction (line) signal TONE OFF;
- \( j_0 \): outgoing junction (line) signal TONE OFF;
- \( d \): (forward) multifrequency code (register signal) DIGIT RECOGNIZED.
- \( \bar{d} \): (forward) multifrequency code (register signal) NO DIGIT RECOGNIZED ("NO TONE").
- \( b \): backward multifrequency code (register signal) DIGIT RECOGNIZED.
- \( \bar{b} \): backward multifrequency code (register signal) NO DIGIT RECOGNIZED ("NO TONE").

(b) Set \( m_{31} = \{ \bar{t}_i \mid i=1 \to 11 \} \)

where \( \bar{t}_i \) = expiry of the corresponding supervisory timer \( T_i \).

2.2 The output signals from the call-handling processes are defined pictorially, as explained in the following Section 3.

3. KEY TO PICTORIAL ELEMENTS IN THE STATE PICTURES

- **ICJ** = interface with incoming junction (ICJ), with backward line signal TONE ON.
- **ICJ** = interface with ICJ, with backward line signal TONE OFF.
- **OGJ** = interface with outgoing junction (OGJ), with forward line signal TONE ON.
- **OGJ** = interface with OGJ, with forward line signal TONE OFF.
- **MFC** = multifrequency code (MFC) sender, transmitting register signal A6.
- **MFC** = multifrequency code (MFC) receiver. (For signal conditions \( d, \bar{d}, b \) and \( \bar{b} \), see section 2.1 above)
- **PATH** = path connected, whereas \( \longrightarrow \) = path reserved but not connected.
- **T_i** = supervisory timer \( T_i \) is running.

Note: Tables 2, 3, 4, and 5 of CCITT Rec. Q.361 should be consulted for the explanation of the meaning of each register signal.
4. SPECIFICATION OF THE R2–R2 TRANSIT CALL-HANDLING PROCESS