DATE: July 23, 1981

TO: R & D Personnel

FROM: Terminal Services Development Group

SUBJECT: The Terminal Services Umbrella Document

REFERENCE: None

KEYWORDS: Terminal, TSD, STI, STICCI, Charter, Status

ABSTRACT

This document is to announce the official existence of the Terminal Services Development Group, formerly part of the Standard Terminal and Intelligent Controller Interface (STICCI) Group. Inside you will find:

- The charter and goals of the Terminal Services Development Group.
- What we see to be our role in Engineering.
- What our current activities are.
- What we expect our future activities to be.
- Where to find detailed information about our various activities.

This document should be read by anyone who needs basic information about the Terminal Services Development Group's purpose and activities.

1 Introduction

This document defines the charter of the Terminal Services Development Group and describes its activities.

The Terminal Services Development Group is a section in the Communications Software Development Group. The Terminal Services Development group has had several different names in the past, such as

- Standard Terminal Interface (STI) group
- Terminal Handling Enhancement (THE) group
- Standard Terminal and Intelligent Communications Controller Interface (STICCI) group

Assumptions might have been made as to what the group's responsibilities were under its former names. The charter section of this document will clearly define the scope of the group under its new name.

Each activity is described in the activities section. (Note that a project is an activity, or part of an activity, but an activity is not necessarily a project.) An activity description consists of a brief statement of what the activity will accomplish, a pointer to any detailed publications concerning the activity, and its status as of the last revision of this document. A complete list of all publications produced by the Terminal Services Development Group is in Appendix B.

This document will be updated periodically with basic information, such as activity descriptions, pertaining to the Terminal Services Development Group.

2 What is the Terminal Services Development Group?

2.1 Charter

The scope of the Terminal Services Development Group is the definition and development of services which constitute the end-to-end terminal system.

One "end" of the terminal system is a program (system program or application); the other "end" is a terminal which is an interface to a human user. The "services" the Terminal Services Development Group will provide are intended to ease the task of software engineers who wish to develop applications that take advantage of the sophisticated capabilities of terminals. These services will include resource management and provision of presentation level functionality. These services are to be provided within a system of distributed processors of varying computing and memory capacity.

Our charter includes, but is not necessarily limited to:

- Terminal system definition.
  We must decide:
  1. What services should exist for terminal support?
  2. How can those services best be provided in the distributed environments of the near future?
- Development of those components of the system that are unique to terminal support.
- Being a focus for terminal system issues.
  There is considerable overlap between our charter and the responsibilities of other groups within Prime, including communications hardware development, terminal hardware development, Primos development, network development, and many others. New products that are to be components of the end-to-end terminal system must be developed in the context of the end-to-end terminal system.

We expect to make use of the new I/O framework being developed by the Logical I/O (LIO) project within the Operating Systems Development group. Although we intend terminal services to apply primarily to terminals, we will of course work within the context of LIO's device independence, wherein the "terminal" may in fact be a disk file or another program.

We expect to be users (not builders) of general transport and session management facilities which will make distributed terminal services (as well as other distributed services) possible. We

expect and welcome architectural work in this area by the Distributed Systems Architecture Group. We have a strong commitment to making the terminal services we define work within a future standard Prime Distributed Architecture as it is developed by DSAG.

There are some elements of a terminal system that might appear to be part of our charter but that we feel are not. Some of these are:

- Ownership of the long term intelligent communication controller strategy.
  The Terminal Services Development Group will be responsible for definition of terminal services to be performed by the new controllers. However, currently envisioned applications of the communications controllers extend far beyond just terminal support. A charter for the group which will manage communications controller issues is being developed separately.

- Ownership of the long term I/O bus communication strategy.
  Because the first usage of the Ring 0 Inter-Process Queuing and Notification Mechanism has been terminal support on the LYNX, we have had temporary custody of the issue. The Intelligent Communications Controllers and Networks sections of the Communications Group, the Distributed Systems Architecture Group, and the New System Project are among the groups which will be providing the complete solution to this issue.

- Definition of terminal products and development of system software which runs inside those terminals.
  This is the responsibility of the Terminals section of the Peripheral Development Group. We have a responsibility to incorporate the terminals they produce into the end-to-end terminal system we will define. Thus we have a clear responsibility to state our requirements for terminals as input to the development of terminals.

2.2 Contributions

Our major contributions to the corporation will be:

- An architecture for an end-to-end terminal system.

- Software that implements the end-to-end terminal system architecture.

- Requirements for other development efforts based on the end-to-end terminal system architecture.

3 What is the Terminal Services Development Group doing?

The current Terminal Services Development Group activities fall into three categories:

- Architectural Development
- Standard Terminal Services
- Non-standard Terminal Services

The activities are summarized below.

3.1 Architectural Development

The Terminal Services Development Group will provide a design model and architectural standards for designers and implementors on the Terminal Services Development projects.

Documentation: PE-TI-837, "Speaking of Architectures...".

Status: Not yet started.

3.2 Standard Terminal Services

A standard terminal interface was the original goal of the Terminal Services Development Group. It is still what most people think of when they think of what the group will provide. Activities in this category will standardize the applications view of terminals to the extent that this is possible and desirable. The standard interfaces will drive any suitable terminal without a change in the applications program, regardless of how functionality is implemented in the terminal.

The first candidates for standardization are the three obvious types of terminals and terminal service: page/scroll mode, forms mode, and graphics mode. Once the basic terminal services for these terminals are in place then new standard services will be defined, e.g., window handling, et. al.

3.2.1 Terminal-Process Binding

This activity will study the concept of dynamic terminal-process binding under Primos. This concept means that terminal lines will no longer be statically connected to a particular process, but will be connected to any available process on demand. Implementation of dynamic terminal-process binding is intended to

- provide a uniform interface for terminals, regardless of type, connection, or use
- allow many more terminals than there are processes to be connected to a system
- provide a better way to support terminals which cannot be statically connected

Documentation: PE-TI-786, "Terminal-Process Binding Prototype Project"; PE-TI-794, "Design for the Terminal-Process Binding Prototype".

Status: This activity is currently on hold as a result of higher priority activities. Its scope may change as a result of the logical I/O effort underway in the Operating Systems Development Group.

3.2.2 Canonical Terminal

This activity defines a model for terminal functionality (the "canonical terminal") as seen by user applications programs. The functionality covered in this activity is that of scroll class (teletype, one-dimensional) and page class (video, two-dimensional). The canonical terminal will allow applications programs to be unconcerned about how the terminal is connected and how a particular terminal implements a particular function.

Documentation: PE-TI-844, "Basic Canonical Terminal Definition Project"; PE-TI-847, Rev. 1, "Canonical Terminal Requirements".

Status: A project definition and a requirements list have been produced. This activity is staffed and work is progressing.

3.2.3 Forms Terminal Project

This project deals with support of the applications program interface to "forms mode" terminals. "Forms mode" terminals are essentially "page-class" terminals with field definition capability.

This project is being done to support the Transaction Processing system. Accordingly, the Transaction Processing Group is providing the main requirements input to the project.

Documentation: PE-TI-519, "Forms Mode Terminal Issues"; PE-TI-520, "Forms Mode Terminal Software".

Status: This activity is staffed and work is progressing.

3.2.4 Graphics Terminal Project

This project is defined but not yet funded. Its purpose is to provide a standard applications interface for graphics terminals.

Documentation: Not yet available.

Status: This project is not yet funded.

3.3 Non-standard Terminal Services

The Terminal Services Development Group will provide a mechanism to preserve current user products and a development bridge to the new terminal services.

All current hardware support activities will implement the current gate-level (TYPERS) interface. The Terminal Services Development Group thus has no need to provide interface support. On the other hand, the Terminal Services Development Group will provide consulting services to the Intelligent Communications Controller Interface development group for the HAWK/FALCON protocol development effort. The purpose is to steer that effort in the proper direction for future terminal support.

3.3.1 Current Terminal Services

The Terminal Services Development Group will maintain only the present gate-level ("TYPERS") terminal services. (Support and emulation of IBM terminals has its own development groups.) Current activity consists of fixing bugs in the asynchronous terminal support logic.

Documentation: Some documentation for the asynchronous support is contained in the PRIMOS Revision documentation. More is contained in the document for the LYNX asynchronous support activity under the Intelligent Communications Controller Support Project.

Status: One person is charged with fixing terminal support bugs as they appear.

Appendix

History of the Group

This appendix contains a brief history of the Terminal Services Development Group. It includes notes on the various groups that in the past have gone under the name "STI" and its variants. This section will be updated as necessary.

Late 1978

The Standard Terminal Interface project was started by Jeff Flowers and Attilio Cecchin of the Primos Development group. The scope of the original project was making it easier to implement and integrate drivers for new kinds of terminal connections. (ELF, remote login and DPTX had just gone through that painful process.) The Primos Development group started the project, realizing that it should also involve the Communications Software group. At the time, manpower resources were not available in the Communication Software group. The Primos Development group proceeded anyway, realizing that the need for such a project was becoming more and more important. Jeff and Attilio held several meetings and produced a functional spec (PE-TI-551).

Mid 1979

The scope of the project grew as people decided there were just too many things wrong with Primos's terminal handling. Participation by the Communications Software group was now considered necessary. STI was renamed the Terminal Handling Enhancement (THE) project to be jointly run by the Primos Development group (Jeff Flowers) and the Communications Software group (Carol Hannauer) since the problem overlapped both areas. An expanded list of goals was informally circulated.

Late 1979 to mid 1980

The name reverted to STI (nobody liked THE). It was wholly absorbed by the Communications Software group and given a full time staff of Dick Wolfson (on loan from Advanced Systems Development) and Evelyn Tate with part time or occasional help from a handful of others. STI published and had reviewed a formal list of goals (PE-TI-715). It studied some issues in depth (especially terminal-process binding, network virtual terminals, scroll mode terminal support, forms mode terminal support, and new controller strategy), and produced a study report (unpublished) in August 1980.

The STI group never had enough resources to handle the whole problem concurrently, and had difficulty focusing on a single issue when so many were so pressing. The group did accrue a lot of useful information and got a good idea of the scope and magnitude of the problem.

August 1980 to April 1981

The STI group was rechartered to include integration of and Primos support for the new intelligent communications controllers, as well as the original STI goals. The new name STICCI was an acronym for Standard Terminal and Intelligent Communications Controller Interface. The new team originally consisted of Ted Gibson, Scott Sminkey and Evelyn Tate. Dick Munroe came on board in September, spending a lot of time initially on the WREN design. John Howell (of the U.K. TP group) began looking at the new FORMS interface from an STI perspective in October.

A prototype effort to do dynamic terminal-process binding was undertaken by Scott in September. Two project documents were produced (PE-TI-786 and PE-TI-794). The project was suspended in November when support for the LYNX communications controller became critical. The Canonical Terminal project was initiated by Ev in October; some project documents (PE-TI-844 and PE-TI-847) were published in March 1981. Serious discussions about the group's charter began in April.

April-July 1981

The group decided that the two functions of STICCI -- terminal handling and communications controller strategy -- could be better handled separately. After a search for a more descriptive name, the "STI part" of STICCI was rechartered as the Terminal Services Development Group with the same members as before. This umbrella document containing the Terminal Services Development charter was first published in July. (A charter for the group which will manage communications controller issues is being developed separately.)

Appendix

Publications

1. Standard Terminal Interface, PE-TI-551, by Cecchin and Flowers, 5-Dec-78

2. Standard Terminal Interface Goals, PE-TI-715, by Scelza, Tate, Wolfson, 13-Dec-79

3. Standard Terminal Interface Project Report (Preliminary Draft), unpublished, by Tate, Wolfson, 14-Aug-80

4. Terminal-Process Binding Prototype Project, PE-TI-786, by Tate and Sminkey, 6-Nov-80

5. Design for the Terminal-Process Binding Prototype, PE-TI-794, by Sminkey, 20-Nov-80

6. The WREN Device Control Stream: What ARE we trying to solve, Memo (file id: DM_002/2), by Munroe, 26-Jan-81

7. Speaking of Architectures..., PE-TI-837, by Munroe and Tate, 5-Mar-81

8. Basic Canonical Terminal Definition Project, PE-TI-844, by Tate, 19-Mar-81

9. Canonical Terminal Requirements, PE-TI-847, Rev. 1, by Tate, 16-Jul-81

10. Forms Mode Terminal Issues, PE-TI-519, by Howell, 2-Jul-81.

11. Forms Mode Terminal Software, PE-TI-520, by Howell, 2-Jul-81.