MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0299493 6

ORNL-TM-10771

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Standards and Conventions for TOPS Use of PSL/PSA

M. M. Stevens
S. S. Stevens
P. T. Singley

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A05 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-10771

Energy Division

STANDARDS AND CONVENTIONS FOR TOPS USE OF PSL/PSA

M. M. Stevens
S. S. Stevens
P. T. Singley

Date Published—December 1988

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
U.S. DEPARTMENT OF ENERGY
under contract No. DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0299493 6

TABLE OF CONTENTS

	Page
ABSTRACT	vii
1. STANDARDS AND CONVENTIONS REQUIREMENTS	1
1.1 GENERAL REQUIREMENTS	1
1.2 PURPOSE OF STANDARDS AND CONVENTIONS	1
1.3 METHODOLOGY	1
2. ROLES AND RESPONSIBILITIES	3
2.1 MOVEMENT OF HOUSEHOLD GOODS	3
2.2 MANAGEMENT OF THE TOPS PROJECT	3
2.3 TOPS PROJECT RESPONSIBILITIES AT ORNL	3
2.4 TOPS SYSTEM OPERATION	4
3. MODELING TOPS WITH PSL	5
3.1 GENERAL USE OF PSL/PSA	5
3.1.1 System Boundaries	5
3.1.2 Less Formal Information in PSL	5
3.2 USE OF PSL OBJECTS AND RELATIONSHIPS	6
3.2.1 High-level Modeling	7
3.2.1.1 Level 0 Modeling	7
3.2.1.2 Level 1 Modeling	7
3.2.2 Intermediate-level Modeling	9
3.2.3 Low-level Modeling	12
3.2.3.1 Complements of Dual-operand Relationships	12
3.2.3.2 Objects and Relationships at Low Levels	13
4. NAMING CONVENTIONS	19
4.1 INTRODUCTION	19
4.2 NAMING CONVENTIONS	19
4.2.1 PSL Naming Conventions	19
4.2.2 Forming User-defined Names	20
4.2.3 Abbreviations and Acronyms in Names	20
4.2.4 Duplicate Names	20
5. REPORTS AND DOCUMENTATION	23
5.1 INTRODUCTION	23
5.2 PSA REPORTS FOR MODULE DESIGN	23
5.2.1 Structure	24
5.2.1.1 Structure Report	24
5.2.1.2 Extended Picture Report	24
5.2.1.3 Picture Report	25
5.2.2 Data Flow	26
5.2.2.1 Picture Report	26
5.2.2.2 Extended Picture Report	26
5.2.2.3 Activity Flow Diagram	26
5.2.2.4 Process Summary	27
5.2.2.5 Function Flow Data Diagram	27

Page

5.2.2.6 Data Activity Interaction Report 28
5.2.3 Control and Dynamics 29
5.2.3.1 Process Chain Report 29
5.2.3.2 Dynamic Interaction Report 30
5.2.4 Properties and Characteristics 30
5.2.4.1 Data Base Summary 30
5.2.4.2 Dictionary Report 31
5.2.4.3 Punch Comment Entry Report 31
5.2.4.4 Attribute Report 31
5.2.5 Data Base Integrity 32
5.2.5.1 Formatted Problem Statement 32
5.2.5.2 Name Selection Report 32
5.2.5.3 Name List Report 33
5.3 PSA REPORTS FOR DESIGN DOCUMENTATION 33

6. PSA DATA BASE INTEGRITY 37
6.1 SECURITY AND ACCESS PROTECTION 37
6.1.1 Access Protection 37
6.1.2 Procedural Integrity 37
6.1.3 Logical Integrity 37
6.2 BACKUP STRATEGIES 37
6.2.1 Project Computer Procedures 37
6.2.2 Personal File Backup Strategies 38

Appendix A. TOPS ATTRIBUTE DEFINITIONS A-1
Appendix B. ACRONYMS AND ABBREVIATIONS USED IN TOPS B-1
Appendix C. PSA DATA BASE MAINTENANCE C-1
Appendix D. WORKING NOTES AND MEMOS ABOUT PSL/PSA USE D-1

ABSTRACT

The Transportation Operational Personal Property Standard System (TOPS) is an automated integrated information management system developed to help administer the personal property transportation program of the U.S. Department of Defense.

Analysis and design information about the system is compiled in a computer data base using the software engineering tool PSL/PSA (Problem Statement Language/Problem Statement Analyzer). This document describes the standards and conventions used in the PSL/PSA representation of the TOPS software system.

1. STANDARDS AND CONVENTIONS REQUIREMENTS

1.1 GENERAL REQUIREMENTS

The Transportation Operational Personal Property Standard System (TOPS) is an integrated information management system assisting personnel at Personal Property Shipping Offices (PPSOs) of the four branches of the armed services with record keeping about the shipment of household goods of members of the Department of Defense. On completion, TOPS will consist of a network of distributed computer hardware, linked by a complex software system. This document describes standards and conventions to be applied to the body of design and analysis information about TOPS compiled with the software engineering tool PSL/PSA (Problem Statement Language/Problem Statement Analyzer).*

The history of large software projects illustrates that the specification of software systems has been an extremely troublesome area, and computer software literature abounds with illustrations of the inadequacies of English as a specification language. In an effort to avoid ambiguity, the TOPS project team at the Oak Ridge National Laboratory (ORNL) has selected PSL/PSA as the principal specification tool and medium for TOPS analysis and program design. Because maintenance and modification of the TOPS system will be carried out by a separate agency and not by ORNL, it is also particularly important that the ORNL developers' corporate memory of TOPS development be carefully described in a systematic and effective set of program documentation.

Because of these two stringent requirements, it is TOPS project policy that all TOPS programming be preceded by careful and accurate PSL representation of the modules to be coded and that all subsequent program changes first be made in the PSL representation. Thus, the project PSA data base is always maintained as an accurate and complete description and specification of TOPS software.

1.2 PURPOSE OF STANDARDS AND CONVENTIONS

This document contains descriptions of the PSL conventions to be used for the TOPS project and the standards that have been set by the TOPS project team for the PSL representation of the TOPS software system. In addition, supporting information is included to allow anyone generally familiar with PSL/PSA to utilize the formal TOPS PSA data base to understand the TOPS system.

1.3 METHODOLOGY

TOPS development follows a modified Yourdon-DeMarco methodology with extensions to allow better modeling of control structures. In general, analysis and design proceed top-down, from very general, abstracted overview through precise, detailed specification to actual implementation. At each level of detail, visualization of the problem/solution is followed by a

---

*PSL/PSA was developed by ISDOS, Inc., of Ann Arbor, Michigan.

representation using PSL/PSA; the representations, when verified, become program specifications. Verified program module specifications are encoded in a high-level, rapid-prototyping language to produce screens and packages that can be tested by transportation specialists. After satisfactory completion of user tests, TOPS modules will be integrated into a complete system for field prototype tests. Extensive user participation in the entire analysis/specification/prototyping process will help ensure that the TOPS system will meet the needs of its user community.

2. ROLES AND RESPONSIBILITIES

2.1 MOVEMENT OF HOUSEHOLD GOODS

By its charter, the Military Traffic Management Command (MTMC) is the U.S. Department of Defense (DOD) organization that has as part of its responsibility oversight of the movement of household goods of DOD members on official travel. In carrying out that function, MTMC performs various rate-negotiating and contracting functions as well as various other interfacing functions with the commercial household goods transportation industry. In addition, MTMC directs the operation of personal property shipping offices at military installations throughout the continental United States and the rest of the world.

2.2 MANAGEMENT OF THE TOPS PROJECT

The TOPS project was conceived to assist personnel at shipping offices at military installations in their performance of services to members and to help MTMC direct efficient operation of DOD household goods shipping. MTMC has designated a TOPS Project Manager to be responsible for the development and implementation of the TOPS project. A TOPS Steering Committee composed of members of the four military services is responsible for policy decisions relating to all-service implementation and use of TOPS. A TOPS Joint Development Team, also composed of members of the four services, is responsible for functional completeness and integrity of TOPS. Under an interagency support agreement, ORNL is responsible for TOPS software system analysis, design, and implementation; for hardware technical specification; and for initial system testing. In addition, ORNL will provide technical assistance throughout the initial field tests of prototype TOPS systems and technical advice during system deployment.

2.3 TOPS PROJECT RESPONSIBILITIES AT ORNL

The structure of the TOPS project at ORNL includes both project-wide tasking and topic-specific tasking. Areas of responsibility are as follows:

<u>Project-wide functions</u>

Project manager
Software coordinator
Database designer
Operating system administrator
PSA database administrators
    Logical gatekeeper
    Physical gatekeeper

<u>Topic-specific tasks</u>

Counseling module
Outbound processing module

Inbound processing module
Nontemporary storage module
Quality assurance module
Switching site processing

2.4 TOPS SYSTEM OPERATION

In the final stages of software system design, ORNL will begin to transmit PSA data base information to the TOPS Maintenance and Modification Agency, with the expectation that system fielding, start-up, and initial operation will be coordinated with that agency. Furthermore, all system design information will be shared with that agency, and responsibility for system maintenance and operation will be transferred to it. The ORNL development team has the responsibility of designing and implementing TOPS so as to facilitate maintenance and modification/extension.

3. MODELING TOPS WITH PSL

3.1 GENERAL USE OF PSL/PSA

At least two kinds of modeling are represented in the TOPS project development effort. In an abstract sense, the entire automated data system called TOPS is a model of the real-world system of persons and objects involved in the transportation of household goods of members of DOD; the stored data will be representations of actual names, weights and other properties. This normal level of abstraction and modeling always takes place in data system development. More pragmatically, the ORNL development team will build a model, using PSL/PSA, to state explicitly how the TOPS software will represent the real world. This section of this document describes how the PSL/PSA model will be built and how the semantics and syntax of the Problem Statement Language are to be utilized in the creation and maintenance of the TOPS system.

3.1.1 System Boundaries

Careful definitions of the PSL objects are given in the ISDOS documentation cited in Section 3.2. For the TOPS project, the following supplementary definitions of the objects Input, Output, and Interface are useful at the program module specification level.

- o OUTPUT is data listed on a printer, displayed on a screen, or transmitted electronically.

- o INPUT is data entered at a keyboard or received electronically.

- o INTERFACE is technically, the operator of the equipment; logically, the person or organization with or for whom the operator is interacting.

It follows that information that remains within the PPSO boundary but leaves and re-enters the TOPS computers may pass to or from an 'operator' as the Interface, while information entering or leaving the PPSO boundary should come from or go to a designated logical Interface such as 'Member' or 'MTMC'.

Note that

- o SET is data that remain within the computer system. As such, it may be updated and have a responsible Interface.

3.1.2 Less Formal Information in PSL

In addition to the formal syntax of PSL, the language provides opportunities to incorporate information informally for tasks that primarily involve interpersonal communication. The ORNL TOPS project team will utilize several such constructs:

- o DESCRIPTION is a complete English-word discussion of a PSL object, beginning with a statement of its full name and including enough

information to indicate to a transportation specialist the relevance of the object to the TOPS system. For a Process object type, the description is a general discussion of what the Process does.

- KEYWORD is used by TOPS developers for the logical grouping of PSL objects. 'Module-name' is the name of the TOPS major function (module) being modeled. 'Level-n' is an analyst-assigned level number that identifies the position of the PSL object within the hierarchy of analysis levels. The analysis level number will be used, for example, to aid in formatting readable reports from PSA.

- ATTRIBUTE is used by TOPS developers to define a characteristic of an object and assign a value to that characteristic. In TOPS, all Elements, Inputs, and Outputs have Attributes associated with them. TOPS Attribute definitions are given in Appendix A.

- PROCEDURE is a discussion of how a Process works. The discussion may include statements in English, structured English, the ORACLE query language SQL, or a high-level language such as C or Ada. The procedure should be complete and detailed enough to serve as coding instruction for the implementor of a module.

- RESPONSIBLE-PROBLEM-DEFINER is the TOPS analyst responsible for the module or for a particular object definition.

- SOURCE is the person(s) or document(s) from which the analyst obtained the information included in the PSA data base.

- DERIVATION is a description of the way a Set is built and maintained.

3.2 USE OF PSL OBJECTS AND RELATIONSHIPS

The syntax and structure of PSL are summarized in various documents published by its developers, ISDOS, Inc., of Ann Arbor, MI. See especially ISDOS Ref.# M0174-8, PSL Language Reference Summary. In brief, PSL uses objects and the relationships among them to model a system. The remainder of this section presents information about the specific objects and relationships to be used to model the TOPS system and its domain.

As discussed in Sect. 1, TOPS development will proceed top-down, through several levels of detail. Accordingly, the PSL model will have various levels, and PSL constructs at each level will be chosen to match the type of information to be represented at that level. In general, at each level required constructs will provide information without which the specification is incomplete. In addition, at each level, optional constructs are available for use when more information would be useful in implementing or maintaining a module.

As experience with the TOPS system grows, this compilation of modeling requirements and recommendations is expected to grow and evolve. At every stage of the TOPS life cycle, this document should serve as a guide to the definitive PSA model of the TOPS system.

3.2.1 High-level Modeling

Information in the PSA data base at the highest levels is intended for communication with a general audience and for orientation of persons requiring the detailed information in lower levels. The PSL constructs reflect these requirements.

3.2.1.1 Level 0 Modeling

At the highest level (Level 0), the entire TOPS system can be represented as a single process, interacting with facets of the military and commercial worlds concerned with movement of household goods. Since the primary purpose for using PSL at Level 0 is interpersonal communication, the following PSL constructs are appropriate:

PROCESS
INPUT
OUTPUT
INTERFACE
SET

At Level 0, each object is required to have a Description, the Keywords 'TOPS' and 'Level-0', and a Responsible-problem-definer. Optionally, each object may also have Synonyms defined. The Outputs have the Attribute Medium.

Structurally, the Process, Inputs, Outputs, and Interfaces are decomposed into Subparts, while the Sets are divided into Subsets. In flow relationships, the Process Receives Inputs, Generates Outputs, and Derives or Employs Sets, while the Interfaces Generate Inputs, Receive Outputs, and are Responsible for Sets.

3.2.1.2 Level 1 Modeling

At the next level of analysis and design, Level 1, PSL is used to describe the general organization of TOPS, with emphasis on macromodularity and information flow. The appropriate PSL constructs and their properties and relationships are listed below.

o PROCESS

Required properties:
- DESCRIPTION
- KEYWORD 'Module-name,' 'Level-n'
- RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
- SYNONYM

Structural relationships:
- SUBPARTS ARE Process-name

Flow relationships:
- RECEIVES Input-name
- GENERATES Output-name

EMPLOYS Set-name
DERIVES Set-name

o INPUT

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

Structural relationships:
SUBPARTS ARE Input-name
COLLECTED IN Set-name

Flow relationships:
GENERATED BY Interface-name
RECEIVED BY Process-name

o OUTPUT

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER
ATTRIBUTE Medium

Optional properties:
SYNONYM

Structural relationships:
SUBPARTS ARE Output-name
COLLECTED IN Set-name

Flow relationships:
GENERATED BY Process-name
RECEIVED BY Interface-name

o INTERFACE

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

Structural relationships:
SUBPARTS ARE Interface-name

Flow relationships:
GENERATES Input-name
RECEIVES Output-name
RESPONSIBLE FOR Set-name

o SET

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

Structural relationships:
SUBSETS ARE Set-name
COLLECTION OF {Input/Entity/Output}-name

Flow relationships:
RESPONSIBLE-INTERFACE IS Interface-name
EMPLOYED BY Process-Name
DERIVED BY Process-Name

3.2.2 Intermediate-level Modeling

There may be one or more intermediate levels of PSL modeling, depending on the relative complexity of the portion of the TOPS system that is under consideration. The principal purpose of the intermediate levels is to facilitate communication between transportation specialists and analyst/programmers to ensure the accurate specification of TOPS functions. PSL constructs and their properties that are to be utilized for intermediate-level modeling are listed below.

o PROCESS

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM
PROCEDURE
SOURCE

Structural relationships:
SUBPARTS ARE Process-name

Flow relationships:
RECEIVES Input-name
GENERATES Output-name
EMPLOYS {Input/Set/Entity/Group/Element}-name
DERIVES {Output/Set/Entity/Group/Element}-name
UPDATES {Set/Entity/Group/Element}-name
MAINTAINS Set-name

Control relationships:
COMES BEFORE Process-name
COMES AFTER Process-name
UTILIZES Process-name

- INPUT

  Required properties:
  - DESCRIPTION
  - KEYWORD 'Module-name,' 'Level-n'
  - RESPONSIBLE-PROBLEM-DEFINER

  Optional properties:
  - SYNONYM
  - SOURCE

  Structural relationships:
  - SUBPARTS ARE Input-name
  - CONSISTS OF {Group/Element}-name
  - COLLECTED IN Set-name

  Flow relationships:
  - GENERATED BY Interface-name
  - RECEIVED BY Process-name
  - EMPLOYED BY Process-name

- OUTPUT

  Required properties:
  - DESCRIPTION
  - KEYWORD 'Module-name,' 'Level-n'
  - RESPONSIBLE-PROBLEM-DEFINER
  - ATTRIBUTE Medium

  Optional properties:
  - SYNONYM
  - SOURCE

  Structural relationships:
  - SUBPARTS ARE Output-name
  - CONSISTS OF {Group/Element}-name
  - COLLECTED IN Set-name

  Flow relationships:
  - GENERATED BY Process-name
  - DERIVED BY Process-name
  - RECEIVED BY Interface-name

- INTERFACE

  Required properties:
  - DESCRIPTION
  - KEYWORD 'Module-name,' 'Level-n'
  - RESPONSIBLE-PROBLEM-DEFINER

  Optional properties:
  - SYNONYM

  Structural relationships:
  - SUBPARTS ARE Interface-name

Flow relationships:
  GENERATES Input-name
  RECEIVES Output-name
  RESPONSIBLE FOR Set-name

o SET

Required properties:
  DESCRIPTION
  KEYWORD 'Module-name,' 'Level-n'
  RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
  SYNONYM
  DERIVATION

Structural relationships:
  SUBSETS ARE Set-name
  COLLECTION OF {Input/Output/Entity}-name

Flow relationships:
  EMPLOYED BY Process-name
  DERIVED BY Process-name
  UPDATED BY Process-name
  MAINTAINED BY Process-name
  RESPONSIBLE-INTERFACE IS Interface-name

o ENTITY

Required properties:
  DESCRIPTION
  KEYWORD 'Module-name,' 'Level-n'
  RESPONSIBLE-PROBLEM-DEFINER
  IDENTIFIED BY {Group/Element}-name

Optional properties:
  SYNONYM
  SOURCE

Structural relationships:
  SUBPARTS ARE Entity-name
  CONSISTS OF {Group/Element}-name
  COLLECTED IN Set-name

Flow relationships:
  EMPLOYED BY Process-name
  DERIVED BY Process-name
  UPDATED BY Process-name

o GROUP

Required properties
  DESCRIPTION
  RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
  SYNONYM
  KEYWORD 'Module-name,' 'Level-n'

```
                IDENTIFIES Entity-name

              Structural relationships:
                CONSISTS OF {Group/Element}-names
                CONTAINED IN {Entity/Input/Output/Group}-name

              Flow relationships:
                EMPLOYED BY Process-name
                DERIVED BY Process-name
                UPDATED BY Process-name

o  ELEMENT    Required properties:
                DESCRIPTION
                RESPONSIBLE-PROBLEM-DEFINER
                KEYWORD 'Module-name'

              Optional properties:
                SYNONYM
                ATTRIBUTES (see Appendix A)
                IDENTIFIES Entity-name

              Structural relationships:
                CONTAINED IN {Entity/Input/Output/Group}-name

              Flow relationships:
                EMPLOYED BY Process-name
                DERIVED BY Process-name
                UPDATED BY Process-name
```

3.2.3 Low-level Modeling

Upon completion of the lowest levels of PSL modeling of the TOPS software system, the PSA data base will contain all the information required to specify TOPS program modules. The data base will be the TOPS program specification. It is important to note that the data base can contain both formal, syntactically verifiable information and informal, descriptive information. It is TOPS project policy to utilize formal constructs wherever such objects and relationships are necessary to ensure logical integrity for the software system and to utilize informal informational constructs when interpersonal communication is the primary intent. The PSL objects and relationships described in the following subsection may be used at the lowest levels of analysis and design.

3.2.3.1 Complements of Dual-operand Relationships

A useful feature of PSA is that it automatically adds complementary relationships to the ones defined by the analyst. Because the TOPS analysis technique focuses on functions and operations, it is recommended that such complex relationships as the two-sided, dual-operand USES (--) TO DERIVE (--) be entered from the perspective of the Process. The complement of the relationship is added by PSA to the definition of each of the operands (data objects) involved. For example, when the PSL statements

```
DEFINE PROCESS    Process-name
USES Input-name  TO DERIVE Element-name
```

are entered into the PSA data base, the complementary relationships are automatically generated for the two data constructs. The definition of Input-name will contain the statement,

```
USED BY Process-name  TO DERIVE  Element-name.
```

Similarly, the definition of Element-name will contain the statement,

```
DERIVED BY  Process-name  USING  Input-name.
```

3.2.3.2 Objects and Relationships at Low Levels

```
o  PROCESS    Required properties:
                DESCRIPTION
                KEYWORD 'Module-name,' 'Level-n'
                RESPONSIBLE-PROBLEM-DEFINER

              Optional properties:
                SYNONYM
                PROCEDURE
                SOURCE

              Structural relationships:
                SUBPARTS ARE Process-name

              Flow relationships:
                RECEIVES Input-name
                GENERATES Output-name
                EMPLOYS {Input/Set/Entity/Group/Element}-name
                DERIVES {Output/Set/Entity/Group/Element}-name
                UPDATES {Set/Entity/Group/Element}-name

              Dual-operand flow relationships:
                USES  {data construct}*   TO DERIVE  {data construct}
                USES  {data construct}  TO UPDATE  {data construct}
                USES  {data construct}  TO MAINTAIN Set-name
                ADDS  Entity-name      TO  Set-name
                REMOVES Entity-name  FROM  Set-name
                MODIFIES Entity-name   IN  Set-name
                REFERENCES Entity-name IN  Set-name

              Control relationships:
                COMES BEFORE Process-name
```

---

* {data construct} may be chosen from {Input/Output/Entity/Set/Group/Element}-name. Note that Output-name may not be used as the left operand.

COMES AFTER Process-name
UTILIZES Process-name

System dynamics relationships:
CAUSES Event-name
CHANGES Condition-name
MAKES Condition-name TRUE/FALSE
TRIGGERED BY {Input/Event}-name
TRIGGERED WHEN Condition-name BECOMES TRUE/FALSE

o INPUT

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

Structural relationships:
CONSISTS OF {Group/Element}-name
COLLECTED IN Set-name

Flow relationships:
GENERATED BY Interface-name
RECEIVED BY Process-name
EMPLOYED BY Process-name

System dynamics relationships:
TRIGGERS Process-name
CAUSES Event-name
MAKES Condition-name TRUE/FALSE

Note that dual-operand relationships involving an Input should be included in the relevant Process definition as discussed in Sect. 3.2.3.1.

o OUTPUT

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER
ATTRIBUTE Medium

Optional properties:
SYNONYM

Structural relationships:
CONSISTS OF {Group/Element}-name
COLLECTED IN Set-name

Flow relationships:
GENERATED BY Process-name
DERIVED BY Process-name

RECEIVED BY Interface-name

Note that dual-operand relationships involving an Output should be specified in the relevant Process definition. (See Sect. 3.2.3.1)

o INTERFACE Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

Flow relationships:
GENERATES Input-name
RECEIVES Output-name

o SET Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM
DERIVATION

Structural relationships:
COLLECTION OF {Input/Entity/Output}-name

Flow relationships:
EMPLOYED BY Process-name
DERIVED BY Process-name
UPDATED BY Process-name

Note that dual-operand relationships involving a Set should be specified in the relevant Process definition. For example,

DEFINE PROCESS Process-name;
ADDS Entity-name TO Set-name;

o ENTITY Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

Structural relationships:
CONSISTS OF {Group/Element}-name

COLLECTED IN Set-name

Flow relationships:
EMPLOYED BY Process-name
DERIVED BY Process-name
UPDATED BY Process-name

Note that dual-operand relationships involving an Entity should be specified in the relevant Process definition. (See Sect. 3.2.3.1)

o GROUP

Required properties:
DESCRIPTION
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

Structural relationships:
CONSISTS OF {Group/Element}-name
CONTAINED IN {Entity/Input/Output/Group}-name

Flow relationships:
EMPLOYED BY Process-name
DERIVED BY Process-name
UPDATED BY Process-name

Note that dual-operand relationships should be specified in the relevant Process definition. (See Sect. 3.2.3.1)

o ELEMENT

Required properties:
DESCRIPTION
RESPONSIBLE-PROBLEM-DEFINER
ATTRIBUTES (see Appendix A)

Optional properties:
SYNONYM

Structural relationships:
CONTAINED IN {Entity/Input/Output/Group}-name
VALUES ARE Integer/'String' THRU Integer/'String'

Flow relationships:
EMPLOYED BY Process-name
DERIVED BY Process-name
UPDATED BY Process-name

Note that dual-operand relationships should be specified in the relevant Process definition. (See Sect. 3.2.3.1)

o CONDITION

Required properties:
DESCRIPTION

KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER
TRUE-WHILE
FALSE-WHILE

Optional properties:
SYNONYM

System dynamics relationships:
BECOMING TRUE/FALSE TRIGGERS Process-name
BECOMING TRUE/FALSE CAUSES Event-name
CHANGED BY Process-name
MADE TRUE/FALSE BY {Process/Input/Event}-name

o EVENT

Required properties:
DESCRIPTION
KEYWORD 'Module-name,' 'Level-n'
RESPONSIBLE-PROBLEM-DEFINER

Optional properties:
SYNONYM

System dynamics relationships:
TRIGGERS Process-name
CAUSES Event-name
MAKES Condition-name TRUE/FALSE
CAUSED BY {Process/Input/Event}-name
CAUSED WHEN Condition-name BECOMES TRUE/FALSE

4. NAMING CONVENTIONS

4.1 INTRODUCTION

This section establishes basic principles and conventions regarding the use of names in PSL/PSA in the TOPS project. These principles are designed to ensure consistency and ease of communication among the project team members and between the team members and transportation specialists. The naming conventions are designed to ensure that the PSL/PSA data base contains only unique, unambiguous names and synonyms, and that their manner of usage is clearly defined.

4.2 NAMING CONVENTIONS

This section prescribes guidelines for creating user-defined object names. The guidelines are needed to provide unique identification of objects within the PSL/PSA data base. A systematic approach to naming PSL objects must be followed to avoid conflicts and inconsistencies in the data base. In addition, consistent and clearly identifiable object names ensure traceability of data and minimize problems in data base integration.

4.2.1 PSL Naming Standards

TOPS standards for naming objects are consistent with the overall standards prescribed in PSL/PSA documentation such as the User's Reference Manual (ISDOS Ref# 82A52-0171-7) and the PSL Language Reference Summary (ISDOS M0174-8).

Valid character sets for object names and for strings are listed in the PSL Language Reference Summary. In general, TOPS names consist of uppercase and lowercase letters of the alphabet (A-Z and a-z), digits (0-9), and the hyphen character (-). A name may not begin with a digit or a hyphen.

Object names are no longer than 30 characters and contain no embedded blanks. Object names that consist of more than one word are hyphenated.

PSL reserves some words for its own specific use. Neither these words nor their synonyms, which are listed in the PSL Language Reference Summary, may be used as part of a user-defined name. In the TOPS project, this should not be a problem because PSL-reserved words must appear in all capital letters, and words in TOPS names are generally composed of lowercase letters. In TOPS names, only acronyms appear in all uppercase letters.

4.2.2 Forming User-defined Names

User-defined names for PSL objects are formed according to the following rules and guidelines:

- Object names consist of full English words that clearly describe the central concept of the object. Object names should be easily understood by the end users of the system.

- Object names may have Synonyms that are shorter, less formal names for the object.

- In general, object names are composed of lowercase letters. Only the first letter of the first word in an object name is capitalized. Other words are in lowercase.

- The use of special characters such as &, +, ., and # should be avoided in object names. This guideline is more restrictive than PSL rules.

- Names of Process objects begin with an action verb and describe the function of the process in words easily understood by the end user.

- Names of data objects are in the singular form.

- A data object that is transformed when it passes through a Process is then known by a new name that reflects the change.

4.2.3 Abbreviations and Acronyms in Names

- Abbreviations defined by PSL for its reserved words and synonyms may not appear in user-defined names. More information on PSL reserved words is given in the PSL Language Reference Summary. See also Sect. 4.2.1 of this document.

- Object names may contain abbreviations when there is insufficient space in the 30-character field for the complete name. (Synonyms may be assigned and used as shorter versions of a name.)

- Acronyms commonly understood by the end user may be used in object names. In the TOPS project, the acronym MTMC is an example. Acronyms in object names are composed of uppercase letters.

- Synonyms assigned to an object name may contain approved abbreviations and/or acronyms.

- Action verbs in Process object names may not be abbreviated.

- A list of abbreviations and acronyms used in the TOPS project is given in Appendix B.

4.2.4 Duplicate Names

PSL/PSA does not allow the use of the same name for different object types. Such errors are automatically detected and identified. PSL/PSA cannot, however, associate similar names for the same object. The following procedures will be used to resolve redundancy and ambiguity in names in the PSA data base:

- PSA reports such as Name List and Keyword in Context will be reviewed periodically to identify naming inconsistencies.

- Obvious naming errors, such as misspellings, incorrectly used abbreviations, and the same name in singular and plural form will be modified to conform to the standards and conventions.

- Names that appear to be similar will be checked through their Descriptions to determine whether the names refer to the same object or different objects. The Responsible-problem-definers listed for the object names will be consulted for further clarification.

- Data element names in particular will be reviewed and checked through a PSA Dictionary report, which includes the Description and Synonyms for each element in the data base.

5. REPORTS AND DOCUMENTATION

5.1 INTRODUCTION

The size and complexity of the TOPS software system dictate that the system be completely modular, and the PSL representation of TOPS mirrors that modularity. A working set of PSA reports has been chosen for use on the TOPS project to ensure consistency and uniformity across a range of program modules created by various programmer/analysts. The reports discussed in Sect. 5.2 are a sufficient set to ensure both that the modules are self-consistent and that interaction among the modules is well-understood.

In addition, certain reports will be generated regularly for the entire TOPS project PSA data base; these reports and their uses are also discussed in detail in following sections.

5.2 PSA REPORTS FOR MODULE DESIGN

The utility of PSL as a design tool is realized primarily through the PSA reports available to the programmer/analyst. The following sections describe PSA reports that will be most useful to the designers of TOPS software modules and recommend usage parameters applicable to many situations likely to occur on the TOPS project. The reports are grouped by the aspect of system design for which they are likely to be most useful. Within a group the reports are ordered by currently perceived usefulness. The system aspects and recommended reports are as follows:

Structure Aspect

- Structure Report
- Extended Picture Report
- Picture Report

Data Flow Aspect

- Picture Report
- Extended Picture Report
- Activity Flow Diagram
- Process Summary
- Function Flow Data Diagram
- Data Activity Interaction Report

Control and Dynamics Aspect

- Process Chain Report
- Dynamic Interaction Report

Properties and Characteristics Aspect

- Data Base Summary
- Dictionary Report
- Punch Comment Entry Report

Attribute Report

Data Base Integrity Aspect

Formatted Problem Statement
Name Selection Report
Name List Report

5.2.1 Structure Aspect

The PSA reports that display the hierarchical relationships among similar objects in a data base, the structure of the data base, are especially useful to reveal the organization imposed on (or discerned in) a problem by the analyst. These reports are described in the following paragraphs.

5.2.1.1 Structure Report (str)

The Structure report presents hierarchies of object types associated by specified relationships. The object named as input for the report is considered level one; an object related to it is level two; an object related to a level two is level three. The level-one name and its associated objects are shown in an indented format indicating the hierarchical structure of the association. Each line of the report contains a level number, the object name, the object type, and the relation type.

In addition to its intrinsic worth, the Structure report may be used to generate a name list for input to other reports, which are then presented in the hierarchical order of the Structure report.

Parameters:

- o Relationships to be considered in this report must be specified through the Relations parameter.
- o Object types to be considered may be specified by the Objects parameter.
- o Direction of the hierarchy may be specified as Upwards or Downwards. Downwards is the default.
- o The Levels parameter restricts the number of levels shown in the report.
- o Many Output options exist. See p. 292 of the PSA Reports and Report Command manual, ISDOS MO223-3.

5.2.1.2 Extended Picture Report (ep)

The Extended Picture report displays, for an object named as input, a network of all other objects in the data base interrelated by input-output flow, data derivation, data structure, system structure, or control. The relationships presented for an object are determined by its type and the system aspect parameter(s) specified in the command. The name, object type,

and the object's relationship to the preceding object are shown for each object on the report.

In graphic format, the chosen object is shown in a box at one side of a page, with related objects shown in a column of adjacent boxes. Objects related to these objects are displayed in the next column, with the pattern repeated until no more relationships are found, a loop is encountered, or a specified number of relationships has been traced. Output may spread over several pages. For greatest utility, the Extended Picture report should be focused by parameters on a limited set of system aspects, such as data derivation and flow or system structure.

The graphic format of the Extended Picture report, as well as its content, make it useful to both the analyst and the user. An analyst can trace linkages from an initializing object to the last of its successors or to a particular target to estimate the overall effect on other objects of a proposed change in the initializing object.

Parameters:

- The types of relationships to be considered for structure are specified by means of the system aspect parameters System-structure and Data-structure.

- In addition, relationships may be specified as either Forward (downward), or Backward (upward).

- The Target-name parameter may be used to trace directly from the initializing object to a particular target object.

- When the Name-only parameter is in effect, only the names are printed for the objects shown on the report.

- The Links parameter may be used to specify the maximum number of links, or connections between names, to be displayed in the report.

5.2.1.3 Picture Report (pict)

The Picture report displays all objects immediately related to the named object. Input-output flow, data derivation, data structure, system structure, and system control related to the chosen object are presented graphically. In contrast with the larger context of the Extended Picture report, which traces a path from the named object through many links to an end point, the Picture report displays only those objects within one link of the named object.

The object being pictured is shown in the center of the report, with related objects around it. The displayed relationships vary according to the type of object under consideration and the choice of system aspect parameters. Object name, type, and relationship to the center object are printed for each related object.

The Picture report can produce a graphic representation of all information about a particular object. Because of this completeness, the Picture report is useful to both analysts and users, at various levels of detail. Pictures of selected high-level objects can display the basic

functions of an entire system, while pictures of low-level objects can help an analyst make completeness checks. (See p. 236 of PSA Reports manual)

Parameters:

- o The types of relationships to be displayed for structure are specified through the Data-structure and System-structure parameters.
- o When the Name-only parameter is in effect, only the name is printed for each object shown in the report.

5.2.2 Data Flow Aspect

Some PSA reports are especially useful to display the flow of information within a system. Those data flow reports not already mentioned will be discussed in the following paragraphs, and reports previously discussed will be related to data flow.

5.2.2.1 Picture Report (pict)

The Picture report, discussed in Sect. 5.2.1 in the context of structure display, is also quite useful to provide pictures of data flow associated with particular objects.

Parameters:

- o The relationships to be considered for data flow are specified through the Data-derivation and Input-output-flow parameters.

5.2.2.2 Extended Picture Report (ep)

The Extended Picture report, also discussed in Sect. 5.2.1, can be a useful tool for understanding data flow through a system.

Parameters:

- o Consideration of data flow relationships is specified through the Data-derivation and Input-output-flow parameters.
- o To avoid complex multipage printout, the number of links to be traced can be limited through the Links parameter.
- o The Boundary parameter should be used with Input-output-flow and Data-derivation to prevent wraparound if a system Interface will be encountered. Boundary causes the trace to stop at the Interface, rather than wrap around out of the system and back in again.

5.2.2.3 Activity Flow Diagram (afd)

The Activity Flow Diagram displays a picture of the subprocesses one level below the initiating process and the data flow through them. (Subprocesses are either subparts of a process or processes utilized by the

initial process.) The format of the afd consists of a box containing the initiating-process name centered at the top of a page or section, with subprocesses in boxes beneath it. For each subprocess, data in are listed to the left and data out are listed to the right of the subprocess box.

The afd report can be especially useful if a list of initiating processes is first prepared by a Name Selection or Structure report with punch file output. When Structure report output is used, the resulting afd displays a hierarchy of processes, with data flow shown at each level below the initiating process.

The afd is useful to analysts and users to give an overview of system structure and data flow in selected parts of a system. In addition, the afd is especially useful to the analyst as a means of checking parallel decomposition of functions and data.

Parameters:

- The Attribute parameter causes values of specified attributes to be printed for the initiating process.

5.2.2.4 Process Summary (psum)

The Process Summary report presents the basic functions of a process, as captured in the informal PSL entries Description and Procedure, and the interactions of the process with data objects. In format, psum displays the Description and Procedure, listings of information entering and leaving the process, and information on sets and entities manipulated by the process.

The data objects are identified by name, object type, and relationship to the process.

The psum report is especially useful for communication between the analyst/designer and the user because it presents the text that has been entered to describe a process and then lists the data that the process uses, derives, or changes.

A series of psum reports may be generated in hierarchical order by using the Punched-name option on a Structure report for all or part of a data base to generate an input name list for the psum. Other desirable series of psum reports can be generated similarly through the selection criteria of the Name-select function.

Parameters:

- The Procedure output option must be specified, since it is 'off' by default.

5.2.2.5 Function Flow Data Diagram (ffdd)

The Function Flow Data Diagram report is a graphic representation of the data flow through a process. For each named process, the ffdd presents, from left to right across the page, information entering the process, the process name, and information leaving the process. Information entering the process is determined through the Receives, Employs, Uses-to-derive, and Uses-to-update syntactical relationships. Information leaving the process is determined through the Derives, Generates, and Updates relationships.

The ffdd presents the same sort of information as the Process Summary (psum) report, in a graphic format, and is therefore also useful for communication between analyst and user. However, the psum contains more information and may be a superior tool for the analyst.

The ffdd reports may be ordered by input name lists from Name Selection and Structure reports.

Parameters:

- o Description and/or Procedure may be printed below the process name.
- o Attribute values may be printed above, below, or with the process name.
- o The Updates relationship may be included in the listing of information entering the process.

5.2.2.6 Data Activity Interaction Report (dai)

The Data Activity Interaction report is a two-part report. The first part reports on the interaction between data objects and processes, while the second part reports on interaction among processes. Each part consists of a matrix for primary display of the interactions and a text summary discussion of possible data base incompleteness revealed by the matrix analysis. The dai is described in outline form:

1. Data-activity interaction

    a. A matrix shows the interaction between data objects and activities (processes or interfaces)
    b. The summary contains messages about possible incompleteness, by listing, for example, a process that derives information without employing or using any information.

2. Process interaction

    a. A matrix shows interactions among processes and data passed directly between them.
    b. The summary identifies
        (1) processes that do not use or employ data derived by another process, and
        (2) processes that do not derive or update data used by another process.

The dai is particularly useful to the analyst to display the interactions of data and their use, to display all data required by a process, and as a means to avoid sudden appearance or disappearance of data objects.

Parameters:

- o Output option parameters allow any part of the output to be omitted.

- Other options specify the number of links to be traced, a system aspect, and order of rows and columns.

- Through the Rows parameter for the matrix output, one may choose rows to be processes and columns to be data or vice versa. It is desirable to display more rows than columns. For detailed instructions about use of the dai, see the PSA Reports manual, p. 88.

5.2.3 Control and Dynamics Aspect

Much of TOPS data manipulation is asynchronous transaction processing, and the ORACLE data base management system is expected to provide the essential control capabilities to guarantee data integrity and prevent problems such as deadlock or deadly-embrace failure. However, in instances in which an element of logical control or sequential processing is dictated by policy, it must be captured by the PSL and programmed into the TOPS modules. For those instances, PSA provides reports such as the ones discussed in the following paragraphs to display the control and dynamics information.

5.2.3.1 Process Chain Report (pc)

The Process Chain report presents, for a named object, a graphic display of the network of objects related to it by dynamics relationships such as Triggers, Causes, and Makes-true/false. The network continues until no more such relationships are found, a loop is encountered, or a specified number of links or relationships has been reached. Object types that may be presented in the pc report are those involved in system control and dynamics, such as Process, Event, and Condition.

The pc report gives a picture of the dynamic aspects of a system and can be a useful means for showing control structure to a user. It can be useful to the analyst in checking the sequence of Processes, Events, and Conditions and in finding Processes that are affected by a triggering event or the occurrence of a postulated condition.

Parameters:

- Through output options, the printing of information about Conditions, Events, and Processes may be suppressed, even though the information is used to determine the relationships displayed in the report.

- The Control parameter may be used to include Comes-before/after and Utilizes relationships, which are not included by default.

- The direction of pc defaults to Forward but may be specified to be Backward.

- Output-format options may be used to control printing of boxes, arrows, etc. For details, see the PSA Reports manual, p. 251.

5.2.3.2 Dynamic Interaction Report (di)

The Dynamic Interaction report produces a two-part matrix/analysis output similar to that produced by the Data Activity Interaction (dai) report. The complete report consists of a matrix that depicts control and dynamics relationships for specified objects and a textual analysis of the matrix that points out potential problems of inconsistency and incompleteness.

The information in the di is similar to that in the Process Chain (pc) report. The pc presents a graphic display of the dynamics relationships between processes and events and is more useful in communicating with the user. The di, which is more concise and includes an analysis section, is more useful to the analyst in verifying proper sequencing of events and processes.

Parameters:

- Through specification of parameters, the di report can show system control (Comes-before/after and Utilizes), system dynamics (Events and Conditions), or both.

- Parameters may be set to suppress either the matrix or the analysis output.

5.2.4 Properties and characteristics

Sections 5.2.1-5.2.3 have discussed PSA reports that deal with the structure, data flow, and control relationships among objects that are expressed in PSL syntax to describe a system. PSL also facilitates the capture of informal information that describes the properties and characteristics of the objects, and a set of PSA reports is available to display the information.

5.2.4.1 Data Base Summary (dbs)

The Data Base Summary report presents statistical information about all objects in a data base. The types of objects, the total count of each type, and the count of each type with specified properties are displayed in tabular form. The dbs provides a quick overview of the size, complexity, and degree of completeness of a PSA data base.

Parameters:

- For TOPS standard reports, Synonym, Description, Responsible-problem-definer, Keyword, Attribute, and Source should be specified as output options. Each property appears as a column heading in the table.

- The npcnt option suppresses printing of percentage columns for each property column.

5.2.4.2 Dictionary Report (dict)

The Dictionary report presents information on properties and characteristics of individual objects in the data base. For each name in the initiating request, dict lists the object name and type and then all properties specified through parameters. In the default mode, all output option parameters are 'on'; the analyst must specify properties to be excluded from the report. Useful properties for TOPS Dictionary reports include Synonym, Description, Responsible-problem-definer, Keyword, Attribute, and Source.

By creating a suitable input name list from a Name Selection report and tailoring the requested properties, the analyst can derive, for example, a data element dictionary. In addition, a dict report that includes all object names and their descriptions is a valuable tool in detecting a variety of problems, such as the use of multiple names for the same object.

5.2.4.3 Punch Comment Entry Report (pcom)

The pcom report presents selected comment entries for the objects specified in a request. Output for each object lists the object name and the text comments. The information is useful in communicating to users the informal understanding gained by the analyst during problem investigation. For the user, pcom can be used to produce a listing of descriptions and procedures for selected processes, to be used in conjunction with other reports such as the Activity Flow Diagram (afd).

Pcom is useful to the analyst as a means of retrieving particular comment entries about objects. For easy modification to comments, the punch file from pcom contains entries ready for editing and subsequent input to the Replace-comment (rcom) command.

The pcom report and the Dictionary report both deal with properties and characteristics of objects. In general, pcom seems to be well suited to display information on processes, while dict seems more useful for data objects.

<u>Parameters:</u>

- Since all output options for pcom are 'off' by default, the comments to be printed must be specified. Entries useful to the TOPS project include Description, Procedure, Derivation, True/False-while, and Layout, depending on the object types given as input.

5.2.4.4 Attribute Report (attr)

The Attribute report prints the names and values of specified attributes for the objects chosen for the report. A tabular format lists the object names in rows, with attribute values in columns.

The Attribute report requires two inputs, the object name(s) and the attribute name(s).

Parameters:

- o Column width in the table defaults to 30 characters; 16 characters is sufficient for currently defined TOPS attributes.

- o Use of the option npspv is recommended to suppress the printing of system parameter values.

5.2.5 Data Base Integrity Aspect

Several PSA reports that have specific utility for purposes discussed in preceding sections are also useful as aids in maintaining the integrity and consistency of a PSA data base in general. Some of these generally useful reports will be discussed in the following paragraphs.

5.2.5.1 Formatted Problem Statement (fps)

The fps presents all the information that is stored in the data base about a given object. The report takes the form of a section of PSL, headed by the object definition statement. All stored relationships with other objects are listed as individual PSL statements (including all inferred reciprocal relationships). Information on the object that cannot be stated in PSL syntax, such as a relationship involving an undefined name, is presented as comments.

Usefulness of the fps is greatly extended because it is presented in a form that may be used as input to generate or change the PSA data base. For instance, the data base may be changed by punching the fps output to a file, editing the file, and then (with suitable precautions) using the file as input PSL.

Because fps output can be input PSL, such a file is a suitable archive record of the state of the software system design. It is recommended that an fps be produced periodically, by the TOPS PSA gatekeeper for the entire data base and by individual analysts for their own module data bases.

Parameters:

- o The Margin parameters may be used to vary the format of the fps.

- o The All-statements parameter is used to print, as comments, all PSL statements that could apply to the named object, even though they have not actually been used for that object.

- o The Punch parameter allows the PSL statements generated by the fps to be 'punched' into a designated file for other uses.

5.2.5.2 Name Selection Report (ns)

The Name Selection report lists all names from the data base that meet the specified selection criteria. Names are listed with their associated object types. In addition to its intrinsic value, the ns report has great utility as a tool to produce input name lists for other reports. Selection

criteria, which vary from simple to quite complex, are discussed beginning on p. 207 of the PSA Reports manual.

Parameters:

- o Order of the report may be specified as alphabetic within object type (default), alphabetic by name over all object types, or by the values of specified attributes associated with the objects.

- o The name of the Punch output file, which contains object names only, may be specified. If no name is given, the default punch file is used, and that file may be reused immediately as the default input to another report.

The Structure report, discussed in Paragraph 5.2.1.1, also may be used to generate a name list for input to other reports. The str list is hierarchical, ordered by organization of the processes or data.

5.2.5.3 Name List Report (nl)

The Name List report lists every object defined in the data base, with its object type and any Synonyms that have been defined. One particular use of the list is to find typing errors, and the Name List report is also useful whenever it is desirable to display all the names of objects in the data base.

Parameters:

- o Order of the list defaults to alphabetic by name within object type, but alphabetic over all object types may be specified.

- o The Date-last-changed may be printed for each object.

5.3 PSA REPORTS FOR DESIGN DOCUMENTATION

The following PSA reports will constitute or be incorporated in primary documentation for the TOPS project. In instances in which the PSA report is the primary constituent of a required document, the form of the report will necessarily be closely defined. When the PSA report is primarily for informal information transfer, emphasis will be on readability rather than adherence to form.

Data Base Summary (dbs)

The dbs (parameters: syn desc rpd key attr src) will be provided to display statistical properties of the entire PSA data base being reviewed.

Structure (str)

A Structure report on Processes will be provided as the central review document. The str report displays a hierarchy of the processes as defined by Subparts and Utilizes relationships.

Supplemental str reports on data objects such as Sets, Inputs, Outputs, and Entities will be provided to illustrate data hierarchies and grouping within the TOPS PSA data base.

Formatted Problem Statement (fps)

Formatted Problem Statements on Processes and data include all information in the data base about each object; they will serve as reference information for project documentation. The fps reports will be organized through use of Structure reports as input files in order to facilitate cross referencing.

Dictionary (dict)

Dictionary reports will also serve as reference documentation, especially for Elements, but including other data types as necessary. Dictionary reports on Processes, with only Descriptions printed, will provide a narrative reference for TOPS functions. Dictionary reports will be in alphabetical order and will include Attributes where they are relevant.

Extended Picture (ep)

Extended Picture reports give a useful graphic overview of system organization. The ep report will be used at high and intermediate levels to illustrate relationships and interactions as seen from various aspects of the system. This complex report usually must be restricted in scope to be readable. See Paragraph 5.2.2.2.

Picture (pict)

Picture reports give a detailed graphic view of structure and flow near a single function. Such reports will be used primarily for intermediate level processes with a great deal of data interaction.

Process Summary (psum)

The Process Summary report can include details about a set of processes, including narrative Descriptions and Procedures, and details of data flow. Psum reports are most useful when they are organized by following a Structure report or in conjunction with a Picture report.

Activity Flow Diagram (afd)

The afd is a concise presentation of Process structure one level below the selected object, and the data flow through the subprocesses. The afd is especially useful at intermediate-to-low levels.

Function Flow Data Diagram (ffdd)

The ffdd is a graphic presentation of some of the information available from a psum report. It will be used in circumstances in which conciseness is required in a graphic overview.

Process Chain (pc)

The pc report is a graphic display of the network of objects associated with a Process by System-control and Dynamics relationships. The pc will be used to illustrate required sequences of processes in TOPS.

6. PSA DATA BASE INTEGRITY

6.1 SECURITY AND ACCESS PROTECTION

Provision is made for ensuring both the physical and logical integrity of the TOPS PSA data base. Responsibility for data base integrity is shared by the entire TOPS team, but three individuals have lead roles in data base control.

6.1.1 Access Protection

The operating system manager administers the UNIX password access control system and the file access code system, thereby limiting access to the PSA files to individuals within the controlled TOPS software group who know valid passwords and who are further granted file access by the creators of the files. Provision is made for routine changes to passwords in the system.

6.1.2 Procedural Integrity

A designated procedural gatekeeper has responsibility for devising, testing, and suggesting procedures for interaction with the PSA data base. Keystroke/command sequences designed to minimize the probability of destructive error are established and strongly recommended to PSL/PSA users. Appendix C discusses procedures to be used in maintaining the TOPS PSA data base.

6.1.3 Logical Integrity

The software coordinator for the TOPS project has responsibility for the logical integrity of the TOPS PSA data base, ensuring that the PSL constructions and conventions described in Sects. 3 and 4 of this document are followed and applied consistently to the accepted TOPS model of the household transportation domain.

6.2 BACKUP STRATEGIES

Because both machines and their operators are fallible, the TOPS project has prescribed backup procedures to allow orderly recovery from errors. The procedures take advantage of distributed processing concepts to provide redundant backups.

6.2.1 Project Computer Procedures

The principal TOPS PSA data base and individual designers' subsets of that data base are resident on the TOPS HP9000 project computer. An incremental tape backup of disk files that have been changed during any day is made at midnight of that day, and the daily backup tapes are kept separate from the computer. In addition, a complete image backup of the entire system disk is made each week and kept remote from the computer.

6.2.2 Individual File Backup Procedures

Analyst/designers who utilize microcomputer work stations are encouraged to duplicate their individual data base components and relevant source files on locally readable backup storage media (normally floppy disks). In addition, printed copies of various reports (especially Formatted Problem Statements) are made at critical stages of module design. Such printed reports could be utilized to restore the PSA data base or to transfer it to another machine if the TOPS HP9000 were completely unavailable.

REFERENCES

Pressman, R.S., 1987. Software Engineering: A Practitioner's Approach, McGraw, New York.

Teichroew, D., and E. Hershey 1977. "PSL/PSA: A Computer-Aided Technnique for Structural Documentation and Analysis of Information Processing Systems," IEEE Trans. Software Engineering 3(1), 41-48.

ISDOS, Inc. 1982. Problem Statement Language (PSL)/Problem Statement Analyzer (PSA) User's Reference Manual, ISDOS Ref #M0171, ISDOS, Inc., Ann Arbor, Mich.

ISDOS, Inc. 1983. PSA Reports and Report Commands, ISDOS Ref #M0223-3, ISDOS, Inc., Ann Arbor, Mich.

APPENDIX A

TOPS ATTRIBUTE DEFINITIONS

ATTRIBUTES FOR ELEMENTS

Type — The type of the data element is determined by the characters that it contains. Values for Type are expressed as strings and include

- 'a' alphabetic
- 'c' character (alphanumeric)
- 'n' numeric
- 'd' date

Length — The maximum number of characters allowed for the element. The value is specified as a positive integer.

Fixed-length — A yes or no indication of whether the element is always the same length. Values are the strings 'y' and 'n'.

Validation — The validation methods that are appropriate for the element. Validation types are expressed as strings and include

- 'r' range
- 'd' date
- 'p' pattern
- 'l' list (enumeration)
- 'n' none
- 'u' unique

Mandatory — A yes or no indication of whether a value must be entered for the element. Values are the strings 'y' and 'n'.

DOD-standard — A yes or no indication of whether the element is included in the DOD list of standard data elements. Values are the strings 'y' and 'n'.

Transient — A yes or no indication of whether the element value is

temporary (that is, not actually stored in the data base). Values are the strings 'y' and 'n'.

ATTRIBUTES FOR INPUTS AND OUTPUTS

Medium — The type of input received or output produced. When more than one medium is used, the ultimate type is specified. Values for Medium are expressed as strings.

Inputs may have the values

- 'keyboard'
- 'transmission' (wire)

Outputs may have the values

- 'pre-printed-form'
- 'plain-paper'
- 'transmission' (wire)
- 'display' (screen)

APPENDIX B

ACRONYMS AND ABBREVIATIONS USED IN TOPS

APO	Air post office
BOA	Basic ordering agreement
BPA	Blanket purchasing agreement
CDR	Contract discrepancy report
CERS	Carrier evaluation report system
CONUS	Continental United States
CTT	Cost and transit time
DITY	Do it yourself
DODAAC	Department of Defense activity address code
DPM	Direct procurement method
DTGBL	Domestic TGBL
ETA	Estimated time of arrival
FPO	Fleet post office
GBL	Government bill of lading
GBLOC	Government bill of lading office code
GOC	Government-owned container
HHG	Household goods
HQ	Headquarters
ICC	Interstate Commerce Commission
ITGBL	International TGBL
ITO	Installation transportation officer
JFTR	Joint Federal Travel Regulations
LOI	Letter of intent
MDC	Movement designator code
MH	Mobile home
MI	Middle initial
MTMC	Military Traffic Management Command
MIA	Missing in action
NMF	Navy management fund
NTS	Nontemporary storage
PBP&E	Professional books, papers, and equipment
PCS	Permanent change of station
POD	Port of debarkation
POE	Port of embarkation

POV	Personally owned vehicle
PPSO	Personal property shipping office
QA	Quality assurance
RDD	Required delivery date
RIP	Reduction in payment
RSMO	Regional storage management office
SCAC	Standard carrier alpha code
SIT	Storage in transit
SPLC	Standard point locator code
SSN	Social security number
TAC	Transportation account code
TCN	Transportation control number
TDR	Traffic distribution record
TDY	Temporary duty
TELEX	Automatic teletypewriter exchange service
TGBL	Through government bill of lading
TO	Transportation officer
TWX	Teletypewriter exchange
UB	Unaccompanied baggage

access	accessorial
acct	account or accounting
accum	accumulated
act	action
addl	additional
addr	address
admin	administrative
allow	allowance
altern	alternative
amend	amendment
amt	amount
approp	appropriation
appt	appointment
auth	authorized

avail	available
carr	carrier
cert	certification
chrg	charge
compl	complete
consol	consolidated
constr	constructive
conv	converted, conversion
debark	debarkation
del	delivery
depn	dependent(s)
desc	description
desig	designated
dest	destination
detach	detachment
discrep	discrepancy
dom	domestic
embark	embarkation
entlmnt	entitlement
est	estimate, estimated
eval	evaluation
govt	government
id	identification
info	information
init	initial
instr	instruction
inter	interstate
intl	international
intra	intrastate
max	maximum
mbr	member
mi	miles
mid	middle
min	minimum

mod	modification
msg	message
no	number
ord	order, orders
orig	origin, original
%	percent or percentage
para	paragraph
pro	professional
pro gear	professional 'gear'
pt	point
qtrly	quarterly
recd	received
redist	redistribution
relatn	relationship
rem	remark
rpt	report
sched	schedule
svc	service, serviceable
ship	shipment
sol	solicitation
spon	sponsoring
stg	storage
temp	temporary
ton	tonnage
trans	transaction
transl	translation
unaccom	unaccompanied
unserv	unserviceable
wt	weight
Y/N	yes/no

APPENDIX C

TOPS PSA DATA BASE MAINTENANCE

1. Logical Procedures

When changes are required in a TOPS module, the Responsible-problem-definer for a module will initiate the required modifications to his module. Large or complex modifications will follow the TOPS methodology, beginning with visualizations and continuing with top-down functional decomposition, but analysts are encouraged to make incremental changes that usually result in at most a couple of pages of PSL.

For such incremental changes, analysts will verify the syntax of the proposed change by submitting the input (usually with IPSL) to a copy of the current project PSA data base.

Syntactically correct PSL will be discussed with the logical gatekeeper for consistency with the project data base and consistency with these Standards and Conventions.

Verified PSL will be placed in the TOPS HP9000 common area, and the physical gatekeeper will be notified of the existence of the files.

2. Physical Procedures

Following procedures outlined in working notes in Appendix D, the physical gatekeeper will make changes to the project data base. Changes will be reviewed with the responsible analyst in order to be certain that intended results were achieved.

Suitable fps reports will be maintained by the physical gatekeeper in the Source Code Control System (SCCS) so that before-and-after images of the data base can be reconstructed as needed.

The physical gatekeeper will keep work-station backups of the current data base at least weekly and after each major change.

The TOPS operating system administrator will coordinate system backup procedures to ensure that backup copies of the project data base are made after major changes to the data base.

APPENDIX D

WORKING NOTES AND MEMOS ABOUT PSL/PSA USE

The following pages contain various working notes generated in response to specific situations or questions that have occurred in the use of PSL/PSA on the TOPS project. This section of the ORNL Standards and Conventions is the most dynamic section, and new working notes will be distributed whenever a need is indicated.

PROCEDURES FOR SHARING NAMES OF DATA OBJECTS IN PSA DATA BASES

Setting up files in the common area

The analyst responsible for an individual data base will maintain a current file of the Set, Group, and Element names defined in his or her data base. A copy of that file and a file containing the path to the individual data base will reside in the common area of the TOPS HP9000 computer for easy reference by other members of the TOPS team.

The file of data object names can be generated with the PSA report Name Selection. The file name should be of the form 'names.module,' where 'module' is an abbreviation of the TOPS module described by the individual data base. The analyst responsible for the quality assurance module could use the following commands to produce a file of QA names and store it in the common area.

```
set o=names.qa
ns s='set or group or element'
sys mv names.qa /users/topsetc/names.qa
set o
```

Note that the Set-output command is used to ensure that both the object names and the object types are included in the file. The punch file from Name Selection contains only the names.

The file containing the path to the individual data base can be created through the vi editor. The file name should be in the form 'path.module,' with 'module' the same abbreviation used for the names file. The contents of the file give the actual path to the data base. For example, the file named path.qa might contain

```
    /users/smith/qual.db
or  /users/smith/qualdir/qual.db
```

For the integrated data base, analysis.db, the list of data object names is contained in names.anal, and the path is contained in path.anal.

Using the files

Members of the TOPS software group will be able to scan the list of data object names for a particular module, such as QA, in the file

```
/users/topsetc/names.qa,
```

and then, knowing the path to the QA data base from the file

```
/users/topsetc/path.qa,
```

a team member will be able to retrieve more information on names of interest from the QA data base through PSA reports. For example, the PSA commands

```
set db=/users/smith/qual.db     (from path.qa)
fps n=Origin-GBLOC
```

would produce complete information on a particular element defined in the QA data base.

When the Set-db command has been executed, any PSA report can be run against that data base. For example,

```
dict n=Origin-GBLOC nrpd nsrc
```

would produce a listing of the description, attributes, and synonyms for the element.

Note that when all needed information from another data base has been gathered, the Set-db command must be issued again, specifying the team member's own data base.

Integrity of shared data bases

In general, there should be no problem with integrity of individual data bases because the files created by individuals in their own areas are automatically protected from modification by other individuals. Specifically, the procedures described above are read (retrieve) operations; they write nothing to the data base.

An analyst may choose to make available to team members a stable copy of his or her data base, rather than a working copy, so long as the stable copy is up-to-date.

GUIDELINES FOR DEFINING FORMS IN PSL

Forms such as DD1299 and AF417 may be defined directly as Outputs, which consist of Groups and/or Elements and are derived and generated by a single Process. There is no requirement for deriving an Entity by the form name and then generating an Output by that name in two separate Processes.

The Output object name should begin with the word 'Form' for easy identification of the form outputs.

The Attribute Medium should be included in the definitions of all Outputs to denote the type of output. Values for Medium are the strings

```
'pre-printed-form'
'plain-paper'
'transmission'      (wire)
'display'           (screen)
```

EXAMPLE of form definition in PSL:

```
DEFINE OUTPUT                 Form-AF417;

DESCRIPTION;

     Form AF417 is certification for payment under the Do-it-yourself (DITY)
     program. This form is used by a member of the Air Force at the
     conclusion of a DITY move to claim his incentive payment;

KEYWORDS ARE 'Outbound';
ATTRIBUTES  ARE  Medium  'pre-printed-form';
CONSISTS OF  Member-identification-data,     (a Group)
             Orders-number,                  (an Element)
             Delivery-address,               (a Group)
             Name-of-TMO;                    (an Element)
RECEIVED BY  Member;           (an Interface)
RPD IS  'MStevens';
```

Note that DERIVED BY and GENERATED BY will be inserted by PSA as complements of the information specified in the relevant Process definition.

INPUT OF OBJECT NAMES TO PSA REPORT COMMANDS

Input individual object name

```
n=Member-name     (data)
n=Capture-member-name    (process)
```

Input file of object names

```
f=names.elem
f=CERSnam.proc
```

1. Create file of names through an editor - vi, for example

2. Use Name Selection (ns) punch file output
   a. File name - default is psaname.nam (which is used by other PSA commands as the default input)
      - can specify punch=userfilename
   b. Order of names - default is alpha within object type
      - can specify alpha

3. Use Structure (str) punch file output
   a. File name - specify pn=userfilename
   b. Order of names - hierarchical
      Especially good with fps, psum, ffdd reports

Use of Noempty option on Name Selection (ns) can build a file of names using the noempty parameter to concatenate output from successive ns commands. The parameter works for both print and punch files.

Example:

```
ns s='synonym'   punch=syn.punch
ns s='attribute' punch=syn.punch  noempty
```

The punch file output from both ns commands will be written to the same file, syn.punch.

PSL VERBS USED WITH OBJECTS NAMED 'TABLE' AND 'RECORD'

In general, a process does not DERIVE a whole table (set).

Instead, it DERIVES a record (entity) and
ADDS the record TO the table, or it
MODIFIES a record IN the table.

At intermediate levels,
a process EMPLOYS a table or
UPDATES a table but does not
DERIVE a table.

A process UPDATES a record or
DERIVES a record.

At lower levels,
a process USES a data object TO UPDATE a record or
USES a data object TO DERIVE a record.

At either level,
a process ADDS a record TO a table,
MODIFIES a record IN a table,
REMOVES a record FROM a table, or
REFERENCES a record IN a table.

NEW METHOD FOR DEFINING REPEATING GROUPS IN PSL FOR TOPS

A repeating group is a group or element that occurs more than once in a given set or entity. For example, since a shipment may contain more than one privately owned firearm, the group Firearms-data may occur more than once in the entity Shipment-request-record. Then Firearms-data is said to be a repeating group.

We originally defined an attribute to be used with the groups and elements that may occur more than once in the same table or record as follows:

```
DEFINE ATTRIBUTE        Repeating-group;
DESCRIPTION;
     The attribute Repeating-group is used when multiple instances of an
     element occur in a given set.  The value of the attribute is either a
     positive integer denoting the exact number of instances or a string
     denoting the range (minimum and maximum) of instances.;
KEYWORD     'General';
RPD         'MStevens';
```

Cy Svoboda of Advanced Systems Technology has suggested that we use System-parameters in the CONSISTS OF statement to denote repeating groups in an entity. The PSL/PSA User's Reference Manual (p. 178) states:

> A system-parameter or numerical value may be specified in the CONSISTS statement for sets, inputs, outputs, entities, and groups. It can be used to denote the number of times a given instance of an object occurs in the containing object.

Use of the System-parameter option in CONSISTS and CONTAINED statements is discussed on page 325 in the same manual.

An advantage to using the System-parameter rather than the Attribute method is that the System-parameter for each repeated object appears on the PSA Contents Report, a kind of Structure report especially for data objects. An example of PSL input and the resulting Contents report follows.

PSL Definition:

```
DEFINE ENTITY           Entity-name;
CONSISTS OF
                        3     Group-name,
```

```
One-to-four Element-name-a,
            Element-name-b;
```

Output from PSA Contents Report:

```
(ENTITY)        Entity-name
(GROUP)           Group-name          (3)
(ELEMENT)           Element-name-c
(ELEMENT)           Element-name-d
(ELEMENT)         Element-name-a      (One-to-four)
(ELEMENT)         Element-name-b
```

The numeric and string values are defined by PSL/PSA automatically as System-parameters. In the Formatted Problem Statements for these parameters, a comment entry shows that the parameter applies to a specific Group or Element in a specific Entity.

PSA Formatted Problem Statement:

```
DEFINE SYSTEM-PARAMETER     One-to-four;
 /* Entity-name
     CONSISTS OF One-to-four
                     Element-name-a /*
```

OUTLINE OF PROCEDURES FOR CHANGING THE TOPS ANALYSIS DATA BASE

1. The analyst makes changes to a copy of the analysis data base.

2. When the analyst is satisfied with the change, the logical gatekeeper reviews the results of the proposed change.

3. The analyst modifies the proposed change to meet the requirements of the logical gatekeeper.

4. When the analyst and logical gatekeeper are satisfied with the proposed changes, a copy of the files needed to make the proposed change is passed to the physical gatekeeper.

5. The physical gatekeeper uses these change files as input for the necessary operations to a copy of the current version of the analysis data base.

6. The analyst and physical gatekeeper check the revised copy of the analysis data base to verify that the changes occurred as desired and that information that should have remained in the analysis data base did not disappear.

7. When the physical gatekeeper is sure that the revised copy of the analysis data base is intact, he punches an fps of the revised data base to the file analysis.psl, places a copy of analysis.psl in the SCCS file s.analysis, changes the name of the revised copy of the analysis data base to analysis.db, and backs up a copy of the fps to floppy discs.

OUTLINE FOR PHYSICAL GATEKEEPING PROCEDURES FOR THE TOPS ANALYSIS DATA BASE

I. Maintain the TOPS analysis data base in an area of its own with limited write access and unlimited read access.

   A. The current data base may be copied by an analyst for use in testing the effects of new PSL.
   B. Changes to the data base
      1. Merge new PSL with an fps from the current data base.
      2. Create new data base from merged fps.
      3. Check new against old data base and against proposed changes.
      4. When new data base has been verified to be correct, age original data base with a version number and make the new data base the current data base.

II. Maintain the fps used to generate the data base in a Source Code Control System (SCCS) file in the same area as the data base with the same strictures on access.

   A. Maintain a copy of the fps for the current working data base in an SCCS file. Analysts may copy the fps for their own use.
   B. When there is a new working version of the data base, the fps from the new working data base will be used to make a delta to the SCCS file. The version number of the delta will be the version number assigned to the working data base when it is aged.

III. Keep backups for the data base and fps.

   A. Besides the normal incremental and periodic backups performed by the system manager, a copy of the working data base will be kept on /users/backup under the name analysis.db.
   B. In addition to the incremental and periodic backup performed by the system manager, a copy of the fps for each of the versions of the data base including the working version will be kept on floppy discs.

PSA REPORTS USEFUL TO TOPS

1. dbs syn desc rpd key attr src npcnt

   Data Base Summary - Table showing types of objects,the total number of each type, and how many of them have the properties and characteristics specified in the command. The parameter npcnt suppresses printing of percentages.

2. str n=High-level-process-name rel=part,utls obj=proc -
   nstat (nlns nlvls nots nrts) (pn=procstr.name)

   Structure report - Indented list showing hierarchy of processes beginning with High-level-process-name. Starting with nstat, the parameters listed above suppress printing of summary statistics, line numbers, level numbers, object types, and relation types. The pn parameter sets the name of the punched output file.

3. str n=Set-name rel=part,cltn,csts obj=set,entity,group, -
   element nstat (nlns nlvns nots nrts) (pn=setstr.name)

   Structure report - Indented list showing hierarchy of data objects in Set-name.
   Parameter ssts can be used in the relations list to include the subsets of a Set.

4. ns s='element' (punch=element.name)
   ns s='element and not desc' (punch=nodesc.elem)
   ns s='element and pd="Smith"' (punch=Smith.elem)

   Name Selection report - Produces a list of names that meet the selection criteria. The first command above will list all elements in the data base, the second will list elements that lack descriptions, and the third will list elements with Smith as Responsible-problem-definer.

5. fps f=str.punch.file ndlcc npeof
   fps f=ns.punch.file ndlcc npeof

   Formatted Problem Statement - Complete information, in the form of PSL statements, on processes or data. Using the punch file from a Structure report will produce an fps report in the same order as the str report. Using the punch file from a Name Selection report will produce an fps in alphabetic order by object type. The parameters ndlcc and npeof suppress printing of the Date-last-changed comment and the End-of-file line.

6. dict f=ns.punch.file nkey nrpd nsrc (ndesc nsyn nattr)

   Dictionary report - Information on the properties and characteristics of each of the objects. In the default mode, all output option parameters are on; the user specifies properties to be ignored. The command line above (omitting the parameters in parentheses) should be of use to individual analysts. Beginning with nkey, the parameters suppress keyword, responsible-problem-definer, source, description, synonym, and attribute.

7. psum n=Process-name prcd

   Process Summary - Shows process name, description, procedure, data entering, data leaving, and data manipulated (as OTHER). The psum and ffdd (Function Flow Data Diagram) reports contain similar information; however, psum additionally identifies the types of the data and their relationships to the process, making psum of more use to the analyst. The psum used in conjunction with a Structure (str) report is very effective. The commands

   ```
   str n=Perform-CERS rel=part,utls pn=CERS.name
   psum f=CERS.name prcd
   ```

   will produce a hierarchy of CERS processes and a psum for each one in the same hierarchical order.

8. ffdd n=Process-name (desc prcd)

   Function Flow Data Diagram - Produces a picture of the data entering and leaving the process. The ffdd is a kind of graphic Process Summary (psum) which is useful in communicating with transportation experts; however, the ffdd does not identify data types or their relationships to the process, which makes it less useful to the analyst.

   The ffdd also may be used in conjunction with a Structure (str) report to produce diagrams in hierarchical order.

9. afd n=Process-name

   Activity Flow Diagram - Produces a picture of the processes one level below the process named (through subparts or utilizes) and the data flow through them. No information is given on data object types or relationships, but the graphic output presents a clear and concise picture of data activity through the group of processes.

10. ep n=Set-name ds
    ep n=Intermediate-level-process-name ss links=4

    Extended Picture report - Picture of the network of all objects associated with the named object, in a forward or backward trace. The

relationship types to be included are specified through the System-aspect parameters:

iof	input-output-flow
ss	system-structure
ds	data-structure
ctrl	control
dd	data-derivation

At least one aspect must be specified. The first command line above produces an ep report on Set-name using Data-structure relationships, while the second line produces an ep using System-structure relationships. The links parameter in the second line specifies that only those objects that are within four links of the named process will be displayed.

11. pict n=Process-name

Picture report - picture of all object relationships one link away from the named object. For a process, this includes its parent, its subparts, and all information entering and leaving the process. The System-aspect parameters used for ep may also be used for pict.

12. dai

Data Activity Interaction report - Consists of two parts. The first part reports interaction between data and processes, while the second reports interaction among processes. Each part contains a matrix representation of the interactions and a summary discussion of possible incompleteness, such as a process that derives information without employing or using any information or an input that is not received by any process.

Output-option parameters allow suppression of any part of the output, a choice of data or activities in the matrix rows, and the number of links to be traced.

INTERNAL DISTRIBUTION

1. Kahra Filliz
2. W. Fulkerson
3. C. B. Grillot
4. Glen Harrison
5. J. C. Kolb
6. R. S. Loffman
7. George T. Privon
8. R. M. Rush
9. Frances L. Sexton
10-14. Martha Stevens
15. Scott Stevens
16. J. R. Temple
17. Thomas J. Wilbanks
18. J. C. Worthington
19. ORNL Patent Office
20. Central Research Library
21. Document Reference Section
22-24. Laboratory Records
25. Laboratory Records-Record Copy

EXTERNAL DISTRIBUTION

26. J. J. Cuttica, Vice President of Research and Development, Gas Research Institute, 8600 W. Bryn Mawr Avenue, Chicago, IL 60631

27. J. P. Kalt, Professor of Economics, Kennedy School of Government, Harvard University, 79 John F. Kennedy Street, Cambridge, MA 02138

28. D. E. Morrison, Professor of Sociology, Michigan State University, 201 Berkey Hall, East Lansing, MI 48824-11111

29. R. L. Perrine,Professor, Engineering and Applied Sciences, Civil Engineering Department, Engineering I, Room 2066, University of California, Los Angeles, CA 90024

30-34. Pam Hicks, HQ, MTMC, ATTN: MT-TP, 5611 Columbia Pike, Falls Church, VA 22041-5050

35. Cyril P. Svoboda, Advanced Systems Technology Corporation, 9111 Edmonston Road, Suite 302, Greenbelt, MD 20770

36. Office of Assistant Manager for Energy Research and Development, DOE-ORO, Oak Ridge, TN 37830

37-46. OSTI, U. S. Department of Energy, P. O. Box 62, Oak Ridge, TN 37831