AN APPLICATION OF COMPUTER AIDED REQUIREMENTS ANALYSIS TO A REAL TIME DEEP SPACE SYSTEM

81-2184

A.M. Farny, R.V. Morris, C. Hartsough, E.D. Callender,
Jet Propulsion Laboratory
Pasadena, California
D. Teichroew, and E. Chikofsky
University of Michigan
Ann Arbor, Michigan

A82-10145

Abstract

This paper describes the experience and lessons learned as authors used PSL/PSA as a supporting tool in the description and analysis of the uplink process. The application of PSL/PSA to the description and analysis of the uplink process and the benefits that the uplink process control task obtained using PSL/PSA is reviewed along with the lessons learned by the "tool builders" in this application of PSL/PSA.

1. Introduction

At JPL, the entire procedure of incorporating the requirements and goals of a space flight project into integrated, time ordered sequences of spacecraft commands, is called the uplink process. As spacecraft have become more capable and complex, the complexity and cost of the uplink process has also risen. The uplink process Control Task (UPCT) was created to examine the uplink process and determine ways to improve it. Two tasks included in the UPCT are:

- analysis of existing uplink process control systems, and, based upon that analysis,
- development of improved designs for future systems.

In order to accomplish these objectives it is necessary to capture a description of the uplink process in such a way that the description:

- provides a uniform set of constructs for capturing the information
- provides a means of abstracting data from the totality of available information
- assists the analysis in determining the consistency of individual pieces of information.

Problem Statement Language/Problem Statement Analyzer (PSL/PSA, Ref. 1), developed by the ISDOS Project at the University of Michigan, was chosen as the supporting tool to aid in capture analysis. Based on the experience of applying PSL/PSA, this paper describes the benefits provided to the task, as well as the lessons learned.

The paper is divided into four sections:

The first section is a description of the uplink process, second is a brief description of PSL/PSA. The third section describes the application of PSL/PSA to the description and analysis of the uplink process. This section also discusses the benefits that the UPCT obtained using PSL/PSA. The fourth section discusses the lessons learned by the project analysts as they applied PSL/PSA. Since the tool is in continuing development, these experiences are valuable. They enable the tool builders to improve PSL/PSA and show the importance of the interactions between the solving of an application problem and the particularization of the tools used in the solution of that problem.

2. Definition of the Uplink Process

The objective of current deep space flight projects is to send unmanned spacecraft to remote locations in space to collect and return information not obtainable from Earth. To meet this objective, a spacecraft is designed with a significant degree of internal control to make the observations necessary to collect this information. The types of observations made must be decided upon, prioritized, and integrated into a series of command program sequences. These sequences are developed by people who operation and use the mission. Each sequence directs spacecraft activities for periods from 8 hours to 28 days, depending upon the frequency of activity during the period. Sequences control the collection, storage, and transmission to the ground of the observation data.

The entire procedure of incorporating the requirements and goals of a space flight project into integrated, time ordered sequences of spacecraft commands is called the uplink process.

The Uplink Process begins before spacecraft launch and continues throughout the mission. The process involves:

- determining the requirements for spacecraft and ground activities, collecting individual requests,
- finding a compatible subset of those requests that can be executed by the spacecraft within the resources available to it,
- designing a spacecraft command (program) sequence, simulating the sequence to verify that it is correct,
- implementing the spacecraft sequence into code for the onboard control computers,
- transmitting the sequence to the spacecraft,
- storing it on board until the proper time for its execution, and
- carrying out the sequence.

This entire process is performed for each mission phase. In each mission phase, several sequences are designed and implemented simultaneously, on a waterfall schedule. Outside of this cyclic activity, Project Management provides direction and support in the form of guidelines, control, and authorization.

3. Definition of PSL/PSA

PSL/PSA is a very powerful tool designed to assist the designer/analyst/engineer in the preparation of specifications of an information system. The tool was originally designed to support business applications. However, the tool has been successfully used for a variety of scientific and real time system designs, and can be used in requirements analysis as well as implementation design.

The support provided by the PSL/PSA tool includes the recording and retrieving of information originally stated by one or more individuals. PSL/PSA can, in some sense, be thought of as a very thorough secretary with an ability to connect threads of specially formatted statements and to be able, upon demand, to retrieve, reformat and display information for the user.

The formal language, Problem Statement Language (PSL), allows the user to make statements about specific examples of generic types of objects (PROCESS, INTERFACE, INPUT, OUTPUT for example) and relate one specific instance to other specific instances through formal relationships (INTERFACE A1 GENERATES INPUT Y1). These statements are analyzed by the Problem Statement Analyzer (PSA) which can follow the threads of connected statements and produce reports.

There are a large number of report types produced by PSA. Typical reports include structures of both functions and data, indented listings, HIPO reports, and data flow analysis charts. The user has a high degree of control concerning both the scope and format of the information presented.

The PSA operates in an interactive environment using the facilities of the host operating system and is as system independent as possible. To achieve this independence, the software, including the database management system, is written in FORTRAN IV, and can be installed on any environment which has a FORTRAN IV compiler, integers with 24 or more bits, and sufficient memory for the largest program.

The requirements for a product system should be expressed in an unambiguous machine processable form. PSL has been designed to meet this goal by being able to accurately and completely express all relevant requirements for the logical design of the product system and by having a precise syntax and semantics. A product system is described by listing its subsystems or components, by giving properties, and by stating relationships among the components. In PSL, each component is called an "object". The concepts of objects, names of objects, types of objects and relationships among objects will be illustrated throughout this paper using material from the uplink process Control Task.

The following is a typical text description of a requirement for a new product system:

"A function called the space mission process control takes user objectives for a space mission from ultimate users and returns interpreted observations collected by a spacecraft from domains of interest in space."

The first step in using PSL is to identify the objects in the system being described. This can be done for the above example by capitalizing the nouns in the text description:

"A function called the SPACE MISSION PROCESS CONTROL takes USER OBJECTIVES for a space mission from ULTIMATE USERS and returns INTERPRETED OBSERVATIONS collected from DOMAINS OF INTEREST in space."

Each of the defined objects has a unique name and each of these objects is described in a different context; "user objectives" represents information passing from "ultimate users" to "space mission process control", "interpreted observations" represents information returned to "ultimate users" by the "space mission process control", etc. In effect, each of these objects represents different types or classes of objects. PSL does not allow blanks in the names of objects, and dashes or underscores are normally used to connect names consisting of more than one word.

The next step in using PSL is to identify the relationships among the objects which have been identified. The relationships can be identified by capitalizing the verbs in the example text description below:

"A function called the space mission process control TAKES user objectives for a space mission FROM ultimate users and RETURNS interpreted observations COLLECTED FROM domains of interest in space."

There are a small number of relationships that may be described by PSL. By taking into account the types of objects defined in the above example and the relationship that PSL allows among those objects, the following correspondence between the text description relationships and the PSL relationships can be made:

Text Relationship	PSL Relationship
TAKES	RECEIVES
FROM	GENERATED BY
RETURNS	GENERATES
COLLECTED FROM	GENERATED BY

The descriptions of the system using PSL terminology are:

```
Space_Mission_Process_Control RECEIVES
  User_Objectives

User_Objectives GENERATED BY Ultimate_Users

Space_Mission_Process_Control
  GENERATES Interpreted_Observations

Interpreted_Observations
  GENERATED BY Domains_Of_Interest
```

Taking these objects and relationships, formal PSL statements can be written and entered into a permanent PSA database. Only a few relationships are illustrated here. The language allows 17 types of objects and 66 relationships (Version 5.2).

4. Construction of a PSA Database

To construct a PSA database, data to build the database must be gathered, naming conventions must be established and a semantic map created to effectively interpret PSL semantics into the real world. This section describes these activities as they were applied to the UPCT.

Data Gathering

In order to perform any type of analysis on the description of the uplink process, a PSA database had to be constructed from existing information. Documentation and diagrams were obtained from individuals involved in the actual process. The information was coded in PSL and incorporated into the database. Concurrently, interviews were held with Uplink participants. Data gathered during the interviews was also coded in PSL and added to the database. These two steps provided both a substantial volume of information, and an initial validation of the model. Interviewing was a multi-step activity in which both new information was obtained and previously obtained information modified and corrected. To assist in this process, PSA reports were produced from the database and given to the individual for review during follow up interviews. Once these corrections, modifications and additions were obtained, the PSA database was updated. This iterative process continued until an agreement between the Uplink process participants and PSA analysts was reached. The data gathering activities continue, and will continue, until the end of the task. This continuing effort does not represent a problem to the project because of PSA's separation of the process of capturing from the process of organizing and re-presenting the information.

Naming Conventions

Each object in a PSA database must have an unique name. The number of objects in the database being approximately 4000 by the end of the task, naming conventions are important. Names currently used in the real system were incorporated into the database since the database was created for purposes of consistency checking and analysis. Due to this convention and the PSL 30 character name limitation, conventions were also established for the abbreviation of names.

Semantic Map

To effectively interpret the formal relations traced by PSA, it is necessary for the analysts and Uplink participants to understand the meaning of the PSA keywords in the context of the uplink process. The "usual" meanings of the PSA terms were generally adopted. The only particular modification was the use of EMPLOYS to indicate that information was used to modify a PROCESS, i.e. procedural information.

5. Value of Analysis to the Study of the Uplink Process

A Conceptual Model

To effectively use PSL/PSA, an underlying conceptual model and associated methodology must be used. The need for such a model forced the participants of the UPCT not only to agree on an underlying model for the description of the uplink process but to consistently apply that model as the analysis progressed.

CONSTRUCTS

The model used contained six types of high level constructs. These constructs were:

- System Structure (the boundary between the system and the rest of the world)
- Function Structure (process structure)
- Data Structure
- Data Derivation (the relations between function and data)
- System dynamics and Control
- System Size and Volume

These constructs provided a template for abstracting the information from the totality of existing data describing the system. (Ref. 1)

A good example of how this underlying model forced a more careful analysis of the uplink process is given by the following example on Process/Processor/Capabilities. During analysis of describing certain activities (functions of the uplink process), discussions with individuals involved in the process soon revealed that the properties of PROCESSORs played a vital role in the integrity of the process. The formalism imposed by PSL/PSA assisted the analysts/engineers in seeing the need for this relationship. (How this relationship was constructed is described in Tool Particularization (below).)

Separation of the Process of Capturing Information from Organizing/Presenting Information

The design of PSL/PSA separates the functions of capturing information, organizing information, and presenting information (report and documentation preparation) into separate activities; activities that may be carried out in any order.

In particular, PSL/PSA supports the "journal" approach. Information was added to the database in a random order, as it became available, or arose in discussions. Due to the nature of PSL/PSA, the new information was properly related in the database. In general, the order the data was added to the database is distinct from the order the information is presented in a report. This journal approach supported the iterative and concurrent nature of UPCT discussions and analysis. For example, the inputs and modifications to the database were prepared independently of the order of the material in any report, as a result neither the task or the work habits of the analysts were paced by the tool.

Since PSA reports were only generated upon demand, the analyst had control over the quantity and the scope of the material produced.

Reduction of Clerical Effort

One major benefit of using PSL/PSA was the reduction in clerical effort necessary to manage a large base of information. As of June 1981, the number of objects (functions/processes, data, attributes, etc.) was over 2060. The dictionary for these 2060 objects was over 140 pages in length. Upon demand it could be updated, re-ordered, or re-issued. For the first time in the history of this task, there existed a common dictionary of terms (descriptions of the objects) being used by the different personnel in the uplink process. Without a tool such as PSL/PSA, it is unlikely that a dictionary of this size would be created, or, if created, that it would be continuously updated.

Clerical effort is also reduced by automatic generation of reports and diagrams. Moreover, each report and diagram contained information consistent with the information presented in other reports. The frequency of report generation varied with the needs of the analyst. Usually a report generation cycle varied from 2 to 5 days.

The ease with which changes or updates were made to the database also reduced clerical effort. When information about one object was changed, all other objects linked to it were appropriately and automatically updated.

Support for Refining Information

We have found that there are different levels of information about the uplink process. One level of information is highly formal, in the sense that it is structured in a precise way.

Information of this type can be captured by the current formal relations supported by PSA. Questions about the structure can be asked, contents of a high level data object can be formally determined, and the processes that derive or update a particular data object can be found.

An example of highly formal information is the functional structure of the uplink process.

```
Space_Mission_Process_Control
        Mission_Phase_Process
                Determine_Reqts_System_Act
                Manage_System_Resources
                Program_System_Activities
                Operate_Spacecraft
                Perform_Mainstream_Activities
        Project_Management
```

There are many relationships in the uplink process that are not immediately supported by the tool. Such relationships usually involve properties that are peculiar to the Uplink system. For example, the issue of process/processor capabilities discussed below is an example of such an intermediate level of information.

A lower level is informal information. This information often serves as the connective tissue that allows the analyst to make the transitions from the formal relations handled by PSA to the real world of the product system. A PSL DESCRIPTION is informal, free form text that is associated with each PSA object, and is used to capture the definition of that object. Another example is the PSL PROCEDURE (also free form text) that is associated with a PROCESS and is used to capture the internal procedural details of a process.

As analysis proceeded and information about the system was captured in the database, migration from one level of formality to a higher level of formality was a common occurrence. While searching for a DESCRIPTION of an object, the analyst often found information that could be captured in an intermediate or formal PSL relationship. Realizing that formal or intermediate relationships are not always clear at the first pass, the analyst could capture information informally rather than "risk losing it" in an attempt to find a formal relationship.

Support for Analysis

One of the claims for tools such as PSL/PSA is that it assists the analyst in ensuring that the functional description of the product system is complete and consistent.

In interviews with uplink process participants, reports from the existing database were instrumental in verifying the correctness and completeness of the database, and generally provided a basis for the interview. After trying different reports, a particular subset was selected. This subset consisted of those reports that best reflected the different aspects of the area under discussion. Format of the reports was selected by what was best received by the individual being interviewed. This subset, presented in a hardcopy package, contained the following:

- Picture report showing system structure and control.
- Picture report showing data derivation and flow.
- Function Flow Data Diagram showing data flow in a different format.
- Formatted Problem Statement giving the totality of information known to PSA about the object.
- Structure Report giving an indented listing of the structure of the data involved with the object.
- Function Flow Data Diagram and a Structure Report giving data flow and data structure information for the subparts of the object.
- Dictionary Report for all names presented in the package, including DESCRIPTIONS, SYNONYMS, ATTRIBUTES, and KEYWORDS.

After presenting this package to the individual for his review and comment a modified package is prepared for the next meeting reflecting the changes and/or additions.

Consistency

Consistency is supported by the syntax analyzer of the input processor of PSA. For example, each object must have a unique name. If a particular name is already used to identify one object and an attempt is made to have that name identify an object of a different type, then the inconsistent input is rejected, and an error message provided to the analyst.

Once information is in the database, consistency is supported by assisting the analyst in checking for objects of almost the same name (for possible spelling errors, causing the same object to be entered more than once), and objects that have similar substructure. A NAME-LIST for instance, gives an alphabetic list of all names. When "Corectness" and "Correctness" showed up next to each other as separate items, the analyst was aware of an inconsistency.

Completeness

PSA formally supports completeness checking by providing several analysis reports. One such report was the Process Summary Report. The report contains Comment Entries, text which describes the process and the data entering the leaving the PROCESS. The report also "flags" a PROCESS description as incomplete when that process does not have an input and/or does not produce an output.

Another analysis-type report used was the Data Activity Interaction report. This report generates a matrix showing which data objects and PROCESSes interact with one another. In addition, it generates an analysis of data and PROCESSes that flags instances of incompleteness. For example, if a data item is not derived (it just "magically" appears) and is used by a process, the message "USED BUT NOT DERIVED BY ANY PROCESS" appears.

An informal method of providing completeness for the Uplink database, was by tagging objects with keywords recording the external source of information. In addition, on objects where the analyst could not obtain a description, the keyword 'Undescribed' was used. Usually, the combination of 'Undescribed' and where the analyst got the item name (another keyword) told the story. This technique flagged items that needed descriptions and indicated the need to look for other sources.

6. PSL/PSA Lessons Learned

This section describes some of the lessons learned by the PSL/PSA analysts as they worked with Uplink Process participants. The lessons fall into three categories:

- refinement of information
- importance of being able to particularize a tool (such as PSL/PSA)
- PSA report acceptability

Refinement of Information

The UPCT task has provided insights into how an analyst abstracts, refines, and makes decisions about a problem. Based upon those insights, the analysis methodology emphasized successive refinement, toward both abstraction and detail. Also emphasized, was the importance of taking "a cut at a problem" and then later reorganizing when there is a better understanding.

The Uplink Process was first described through the eyes of the Sequence Team, one of the organizations involved in the process. Hence, activities were described as being performed by organizations and data flow was described as passing from one organization to another. This top level description then had to be redescribed in PSL statements. The first task was to break it into a functionally oriented structure instead of an organizational one. The activities described in the new presentation were defined as PSL PROCESSes and the organizations and individuals involved in performing the activities were defined as PSL PROCESSORs that PERFORMED the PROCESSes. The original description was a 13 box diagram of these major PROCESSes. This was then abstracted to a higher level diagram containing 5 boxes.

This initial refinement of information created a structure or "lattice work" from which the description could be based.

Importance of Tool Particularization

Perhaps the single most important lesson learned during this task was the importance of tool particularization. The PSL/PSA tool had to be "fine tuned" to the peculiarities of the Uplink Process, that is, particularized to the specific problem. This can best be explained through example.

Human PROCESSORs and their experience "properties" played a vital role in the integrity of the Uplink Process, but no PSL relationship between PROCESS and PROCESSOR including their properties is available. To record that a PROCESS' requirement that a PROCESSOR have a specific "property", meant that the basic PSL/PSA tool needed to be "refined".

A set of PSL ATTRIBUTEs was created describing the different areas of experience that a PROCESS may require of its PROCESSOR An associated set of values, SYSTEM-PARAMETERs, was also created to reflect the degree of expertise required in each area. This particularization provided the following set of ATTRIBUTEs and values:

ATTRIBUTEs - "Experience Areas"

Experience_In_Mission
Experience_In_Spacecraft
Experience_In_Instrument
Experience_In_Administration
Experience_In_Clerical

SYSTEM-PARAMETERs - "Degree of Expertise Required"

LOW
MEDIUM
HIGH
EXPERT

A relationship was still needed to show that a requirement existed. The ASSERT statement, a relationship in PSL which shows that one object expects another object to have certain ATTRIBUTEs with certain values, provided the "thread".

The PSL statement then read:

```
     DEFINE PROCESS process-name;
       ASSERT processor-name
                  Experience-Area
Degree-Of-Expertise-Required;
```

Taking the particularized tool, the required values in each area of experience were quickly defined by Uplink Process participants and the existence of PROCESSOR requirements was derived.

The following example, in the case of the activity "Collect_Requests", shows the total result of the particularization in PSL statements.

```
DEFINE PROCESS Collect_Requests;
  PERFORMED BY Experiment_Representative;
  ASSERT Experiment_Representative
         Experience_In_Science HIGH;
```

In other words, the activity Collect_Requests required that the individual it is to be performed by, Experiment_Representative, have a HIGH degree of Experience_In_Science.

This particularization of the tool provided a defined way of recording the peculiarities of the Uplink Process.

PSA Report Acceptability

In presenting PSA reports to interviewed individuals, some general approaches were developed.

- "Feeling out" an individual to judge which reports are more suitable should be done at the beginning of a series of interviews. This result in expediting the interview sequence.
- Straightforward, uncluttered reports (i.e. a Picture report concerned only with system structure) are more likely to prompt the individual into providing more information and corrections.
- Topical, focused reports in the areas of the individual's interest, were best received. Other types of data were confusing.
- A structured, consistent package of PSA reports and brief instructions on how to read them was needed. This package materially assisted the individuals in reading and comprehending the materials.

7. Conclusions

The important issues covered in this paper are (1) the advantages of using a formalism to capture the descriptions of the uplink process and storing them in a database. These advantages include the ease of update of the information and the assistance in determining the consistency and completeness of the design, (2) The necessity for iteration during the design process and the assistance that a tool such as PSL/PSA provides to such iteration.

The major lessons learned by the analysts are (1) the advantages that accrue to the users when a tool such as PSL/PSA can be particularized and (2) the techniques and strategies that can be employed to support iteration of design activities.

References

1. Daniel Teichroew and Earnest A. Hersey III, "PSL/PSA: Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems", IEEE Transactions on Software Engineering, Vol. SE-3, No. 1 (January 1977), pp. 41-48.