TRACER: A TOOL FOR TRACING AND CONTROL OF ENGINEERING REQUIREMENTS

Richard L. Stoller, PhD
Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, California, 91109

ABSTRACT

The Tracing and Control of Engineering Requirements (TRACER) system was developed by JPL to document, trace, and control requirements for any size project. It is a computerized database system designed to support a flexible project requirement document tree that may be partitioned according to level. Individual requirements can be entered at any level and traced from highest source to ultimate implementer. A number of attributes characterize each requirement. A variety of paper and electronic reports is available. The system runs on IBM ATs. TRACER is currently in use for an interplanetary spacecraft project and this paper describes the motivation for TRACER, its purpose, benefit and applicability, provides a summary description of the database system, outlines its hardware and software implementation, and explains its operation. Examples of documents and reports are provided. It is concluded that TRACER can support the requirements documentation and tracing of a full-up project containing more than 25,000 requirements such as an interplanetary spacecraft project.

INTRODUCTION AND OVERVIEW

A complex spacecraft project requires a large set of project documents to define and control requirements for the various elements. In the past, these have generally been handled by manual (paper) methods. Some of the problems with the paper systems have been: incompatibility of requirements written in different areas, difficulty in maintaining current documents, lack of traceability of requirements from high level origination to detailed destination level user, identification of potential impact of changes, and delayed communication of approved changes.

JPL has developed a system to automate the generation, tracing, reviewing and managing of requirements using a computerized data base. It is called the TRACER (Tracing and Control of Engineering Requirements) system. Its purpose is to relieve or eliminate the above described problems.

A preliminary version of TRACER has been used successfully at JPL on the Pathfinder Project. A completely rewritten and very significantly improved version is currently being used on the Mariner Mark II project to document requirements for the CRAF and Cassini interplanetary spacecraft.

The TRACER system supports definition of a list of documents that comprise a flexible project requirements documentation tree organized for each functional area and partitioned according to requirement level (the highest level is the sponsor with each succeeding level identifying more detailed implementation requirements). As currently used at JPL, TRACER accommodates a document tree functionally organized compatible with the JPL Viking/Voyager/Galileo planetary spacecraft requirements system.

This approach has provided a smooth transition from the old to the new requirements system by keeping a documentation tree that is similar to what the implementing groups are accustomed to. Each requirement entered into TRACER is unique, derives from one or more "parent" requirements, and specifies one or more "children" who must respond. Associated with each requirement are a variety of "attributes" which characterize it.

With this requirements architecture, changes to higher level requirements can be easily followed down through the implementing levels. This facilitates rapid impact assessment by the implementing groups and management action to correct or modify changed requirements that have detrimental results (e.g., cost increases or schedule slips). Multiple options for formal and working documents and reports are provided.

The present system is dBASE III+ code, compiled by the clipper compiler (Nantucket Corp.) which runs on IBM ATs with multiple batch input to a single central data base.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

Reports and documents can be output and distributed electronically on a local area network. An improved version having fully interactive and multiple user input/output is planned.

The TRACER system has demonstrated significant improvements over standard paper requirements documentation, can be configured to match "top-down" requirements documentation trees, and has the capability to identify the effects of new and changed requirements for impact assessment and management attention/action.

OTHER WORK

A limited amount af work has been published in the area of general requirements traceability tools. A previous version of TRACER, called the JPL Functional Requirements Tool, was documented in reference 1. That paper goes into considerable detail describing the general requirements documentation problem during the development of an interplanetary spacecraft flight project and explains the approach to the solution used for the development of the Functional Requirements Tool. TRACER is an outgrowth and very significantly improved version of that work.

A reportedly effective proprietary tool is in limited use in the aerospace industry as described in reference 2.

The JPL Space Station Program Office in Reston, Virginia, is working on requirements for an Automated Requirements Management System (ARMS) under the cognizance of G. Giffen and J. Skinner. It will have some characteristics similar to TRACER, but it must be designed to handle a one to two order of magnitude greater number of requirements, and must serve multiple users in different locations. That work is still in a very preliminary stage and will probably not be operational for several years.

There are no other products currently available in the marketplace that this author is aware of that meet the need for general requirements documentation and traceability.

Effort has been expended to develop methods to document and trace software requirements. Reference 3 defines an approach for requirements traceability using an automated tool. One example of a microcomputer-based tracking system developed to support and automate the top-down structured approach to software development reported in reference 4 provides software traceability, requirements testability, and auditing. Developments in specification and tracking tool support are discussed in reference 5. Reference 6 describes structured design concepts. Structured analysis and design is combined with technical management tracking in reference 7.

MOTIVATION FOR TRACER

A Typical Requirements Documentation System

A complex set of project documents are normally generated for a project. The requirements are then handled almost entirely by manual paper methods. One of the key components of a typical spacecraft documentation system is the Functional Requirements Book, which responds to many of the project documents and serves as the basis for development of the spacecraft design requirements.

Problems With A Typical Paper System

Some of the problems with a typical paper documentation system are that requirements written in many different areas with changes occurring as a continuing process, lead to incompatibility. This shows up not only in the documents themselves, but also in project reviews (as well as sometimes in the final design). One area documents (or presents) requirements levied on other areas while the other areas document (or present) the collection of requirements they assume are being levied on them, as well as the requirements they are levying on others; often the common areas do not match.

Also, maintaining the currency of documents is difficult. When a problem exists with one section of a document, it can hold up the approval and release of the entire document. The result is that only the document custodian knows the real status of requirements in his area, because released documents are often obsolete.

Traceability of requirements to the ultimate user is time consuming and often incomplete. There may be long delays before a user discovers that a requirement affecting him has been changed. Sometimes contracts contain references to obsolete requirements.

Finally, there is no simple way to determine which requirements could be easily changed if there were a problem, or which ones are not currently being met.

Purpose of TRACER

The purpose of the TRACER system is to automate the generation, tracing, reviewing, and managing of requirements using a computerized data base.

It is important to understand that the engineering is done the same as before. Communications, negotiations, interactions, iterations, etc., must continue. The TRACER system simply allows the results to be documented in a logical, easily accessible, traceable manner, with one requirement written in only one place, but accessible from many places.

Benefits of TRACER

Advantages of the system are that it:

- Requires succinct and precise identification and treatment of real requirements separate from descriptive prose
- Stimulates communication between engineers early while requirements can be most easily negotiated
- Identifies open links in requirements traceability
- Displays Open/Fault/TBD areas for work
- Creates complete documents with great flexibility
- Includes headers and descriptive material anywhere, any length
- Allows individual approval and release of requirements
- Displays current requirements and status at any linked computer terminal at any time.
- Communicates changes between interfaces
- Assures consistency in documents
- Assures consistency in reviews
- Provides many types of reports
- Provides management visibility

Disadvantage of TRACER

TRACER requires a significant effort during the early stages to accustom the engineers to the rigor of isolating and tracing requirements. One full-time data entry person is common.

Applicability of TRACER

The TRACER System can be used on nearly any sized project, including an in-house interplanetary flight spacecraft project, with many external interfaces. It probably would work well for systems containing more than 25,000 requirements and is constrained primarily by total data storage capability. The largest use of TRACER to date contains more than 6,000 requirements.

TRACER DESCRIPTION

Overview

The TRACER system is an automated mechanism for documenting both descriptive material and requirements. Requirements are linked and traceable; description is generally not (although TRACER allows it).

TRACER requires a list of the documents that comprise the project documentation tree. It is functionally organized similar to the JPL projects Viking/Voyager/Galileo requirements systems, so it is a reasonable transition from that type of all paper system to a computerized traceable system. It is unique in that this documentation system aids the engineering process in the definition and tracing of requirements.

Documents are defined uniquely by the project for each functional area (they can be analogous to traditional JPL project documentation). The documents can be partitioned according to level (sponsor, project, system, subsystem, assembly, etc.). Figure 1 shows the TRACER document tree currently used for the JPL Mariner Mark II CRAF and Cassini Project. Each document has a custodian and a list of approvers. Requirements are organized within these documents according to subject matter.

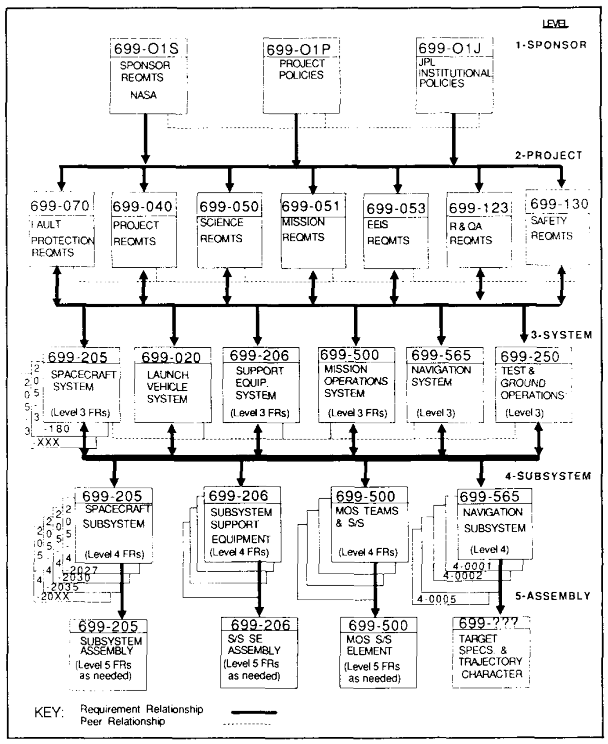

Figure 1. Mariner Mark II Project Requirements TRACER Documentation Tree

Each requirement is stored in a separate record in a database. Each should be a single "shall" statement (multiple statements are allowed by the system but are not generally recommended), and each has a unique identification (ID) number assigned by the TRACER system. It is recommended that requirements be located in a document under a numbered paragraph heading, according to the subject matter. The requirement may have some associated characteristics attached to it.

Requirement Relationships

A requirement derives from a "parent" requirement and is, itself, a requirement expanding or responding to that parent requirement. It may have (unlimited) multiple parents to the degree that it responds to more than one requirement. The parent requirement may be in any document, constrained only by project established rules (enforcement by the TRACER system option afforded by the "System Configuration" menu is not yet operational). Changes of a parent requirement are flagged and presented to the custodian for review and "clearing".

It may be that in some cases a requirement is written in a certain way because the custodian has an agreement with another custodian about another requirement in another document. This dependency can be tracked by specifying the two dependent requirements as "peers". Peers may be linked in any document (constrained only by the project rules) and an unlimited number of peers can be linked. Changes in any peer of a requirement are flagged by the TRACER System and presented to the custodian for review and clearing.

Each requirement specifies one or more "children" who must respond to the requirement. Every requirement (except at the lowest level) must have at least one child. This levying of requirements may continue successively from the highest level, through intermediate levels, to the lowest level, at which point the requirement is satisfied by specifying (as a child) an "implementation" activity such as a contractual agreement, a specification, a design document, or other implementation responsibility, (e.g., the cognizant engineer). Note that a requirement is not considered completely defined until both its parentage and its children (or implementation) have been specified.

A requirement may refer to a table or a figure (i.e., the children must meet the requirements specified in the table or figure). The referenced tables print out automatically at the end of the document, whereas, the figures are manually attached at the end of the document. Tables may be referred to from any document eliminating the need for duplication.

If a document custodian wants to use the exact words of the parent requirement statement, unchanged, as the words for his requirement statement and have the words in his statement automatically updated to mirror the changes of the parent statement, the requirement may be designated as a "pass parent" requirement. The custodian may then either specify a child or an implementation for the requirement. The custodian can be notified when changes to the parent occur, but the changes automatically pass into the requirement statement in the document, without any action on the custodian's part.

Requirements may be levied, according to project rules, generally onto documents at the same level or one level lower. (Going "up" creates confusion and opportunities for loops. Skipping a level down avoids management control and visibility. Both are not recommended.) Parent, peer, and child links may be added or deleted at any time through a TRACER controlled procedure.

All requirements should be considered as negotiated agreements and whether it is the the parent or the child that is the driver often changes as the project develops, so which is the driver is not tracked by TRACER.

A constraint is handled just like a requirement. It is entered in a higher level document, which controls the interface, and one child (who needs it) and another child (who must provide it) are specified.

Requirement Attributes

In addition to the "statement" of a requirement, there are entries that can be used where appropriate to specify other characteristics. They are summarized as follows.

"value" - an alphanumeric requirement (if appropriate), e.g., 5 degrees.

"goal" - an alphanumeric goal if different from the requirement, e.g., 3 degrees.

"capability" - expected performance of the child (for information only if known to be significantly different from the requirement), e.g., 7 degrees.

"reference" - table, figure, or external document that is a part of the requirement.

"comment" - explanation, memorandum, trade study, notes, etc.

"key phrase" - part of the requirement statement contained within a variable (even) number of delimiters. Reports may print the full statement with or without delimiters or only the "key phrase within the delimiters".

This feature is useful for presentations (see figure 6) or succinct work summary lists.

"author" - who wrote this requirement?

"date" - date of last significant change.

"Phase" - what phased delivery does this requirement apply to?

"Mission" - in a multi-mission project, which (if not all) does this requirement apply to?

"(M)eet parent" - does this requirement meet its parent requirement(s): meets, no, ?, etc.

"(A)pproval" - approval status: approved, no, draft, etc.

"(F)irmness" - communicates how easily this requirement could be changed: firm, soft, reviewing, conflicting, etc.

"(E)CR" - is there an engineering change request outstanding against this requirement? What is the number?

"Test codes" - how shall this requirement be verified?

"Flags" - ten undefined one-character flags for any unforeseen needs.

"Fields" - five undefined fifteen-character fields for any unforeseen needs.

Multiple codes may be used for many of the above attributes to allow as fine a characterization of the requirement and its status as desired.

Reports and Document Printout

A wide variety of reports is available containing selected information in multiple formats. The options as shown in figure 2 are explained in the following.

After a report selection is specified, it can be named and saved for repeated use.

The first column allows sections to be selected or rejected. In addition to items explained previously, special interest items include the Table of Contents, which is generated automatically by TRACER from the headings in the document, showing the page numbers. A List of References is automatically generated from the references in the requirements. Parents, peers, and children "linked" to requirements in the document are printed and ordered according to the documents in which they reside. A separate "unlinked" list is created for which the custodian must take action to add the links, e.g., for parents,

```
Report name: Monthly complete dump

(1) REPORT SECTIONS
  _ Cover
  _ Printout record
  _ Approvers
  _ Code key
  _ Scope
  _ Table of contents
  _ List of references
  _ Linked parents
  _ Linked peers
  _ Linked children
  _ Unlinked parents
  _ Unlinked peers
  _ Unlinked children
  _ Key parents
  _ Headers
  _ Requirements
  _ Descriptions
  _ ID index
  _ Tables
  _ Figures
  _ Orphan to parent
  _ Orphan to peer
  _ Orphan to child
  _ Non-complying children

(2) OPERATOR PARAMETERS

  _ To one file: DUMP.TXT
  _ No file
  _ Separate file per doc.
 Path to files _________

60 Page Length
10 Para. no. display width
 0 Left margin

(3) FORMAT OPTIONS

 _ Page no. on each page
 _ Report date on page top
 _ Document heading on pages
 _ Column headings

(4) REQUIREMENT SELECTION

Which requirements:
 _ All requirements
 _ Rqmts changed since mm/dd/yy
 _ Faults
 _ Adhoc search for __________

(5) ADDITIONAL INFORMATION

R P E C
_ _ _ _ Left flag column
_ _ _ _ VGC  (x or c)
_ _ _ _ MAFE (x or c)
_ _ _ _ ID   (x or c)
_ _ _ _ Date (x or c)
_ _ _ _ Document (x or c)
_ _ _ _ Custodian
_ _ _ _ Key phrases
_ _ _ _ Reveal delimiter
' ' ' ' Delimiter
_ _ _ _ Parent IDs
_ / / / Parent statement
_ _ _ _ Peer IDs
_ / / / Peer statement
_ _ _ _ Child IDs
_ _ _ _ This doc. only
_ / / / Child statement
_ _ _ _ Comments
_ _ _ _ Test codes
_ _ _ _ Missions
_ _ _ _ Phase
_ _ _ _ Faults

Change-bar dates:
03/18/88 03/18/88   / /     / /
  / /      / /      / /     / /
  / /      / /      / /     / /
  / /      / /      / /     / /
```

Figure 2. Report Options Selection Sheet

peers, or children which may not have been seen before but which have been levied on the document by custodians of other documents. The ID Index at the end of the document shows the page in the document where each requirement ID is located. Orphan to parents, peers, or children are those requirements in the document that are not linked to requirements in other documents.

Section 2 sets parameters, like line lengths, etc.

Section 3 has options for formatting.

Section 4 allows selection of requirements to be printed. Choices include all, only re-requirements that have changed since a specified date, only those with "faults" ("fault" options selections are shown in figure 3), or requirements satisfying any search key.

```
"x" the faults to test for:
   Children requesting disconnection
   Parentless requirements
   Requirements that do not meet their parent or peer
   Childless requirements
   Requirements under review (firmness = "R")
   TBD in statement, value, or implementation
Display a note stating the fault with each reqmt.
```

Figure 3. Fault Options Selection Sheet

Section 5 of figure 2 gives choices of what additional information is printed with each (R)equirement, (P)arent, p(E)er, or (C)hild.

Finally, four change bar dates can be selected for each of the requirements, parents, peers, and children. All those with change dates later than specified will have a change bar printed in the right-hand margin.

Figure 4 is a sample page from a document showing many of the optional fields. Figure 5 shows similar requirements with a reduced format print selection.

There is also an option for large size print suitable for presentations. This option in combination with a "Selected List" allows one to list headers, requirements from any document in any order, and any comments and the TRACER will print out camera ready viewgraphs. Figure 6 shows a sample viewgraph.

Any of these report types can be output electronically and read from any computer terminal connected to the network.

A "Status Report" provides statistics on the state of all the documents and is useful for management.

```
Page No. 7                        040  Project Requirements
07/28/88
 4          .TEST AND OPERATIONS REQUIREMENTS                              ID

 4.1        ..LAUNCH OPERATIONS

 No power shall be required from the launch vehicle.                       60
                                                     MAFE: ?*R
   Wertz  02/11/87
   Child IDs:   +020        6005  ,  205-3-120     0  , +205-3-250  5923  ,
                +205-3-250  6895  ,

 The spacecraft shall be assembled at SAF and prepared for launch before  5886
 leaving JPL per Galileo ship and shoot philosophy (modular dissassembly
 at Pasadena and reassembly at the Cape for launch).
                                                     MAFE: ?*R
   Draper  03/01/88
   Parent IDs:    01P         1445+ ,
   Child IDs:     250            0  ,

 KSC operations shall be limited to launch readiness activities, launch    209
 vehicle interface verification, and launch countdown tests.
                                                     MAFE:  *F
   Wertz  06/19/87
       Rationale: This concept is known as 'ship and shoot' and is
   intended to minimize the cost of KSC operations by short operations,
   consistent with safety shipping constraints and launch vehicle
   integration requirements.
   Parent IDs:    01P         1445+ ,
   Child IDs:   +205-3-340  6029  ,

 The launch vehicle shall provide water lines to cool the RTGs.            2614
                                                     MAFE:  *R
   Wertz  11/27/87
   Parent IDs:    01P         1445+ ,
   Child IDs:     051-1          0  , +020        6005  ,  206          0  ,
```

Figure 4. Sample Document Page With Many Options Selected

Page No. 4 040 Project Requirements
07/28/88

3.2 ..FAULT PROTECTION	ID
Fault Protection shall ensure that no error or component failure will compromise minimum mission science objectives at the comet or at Saturn. Change: Spacecraft description in the FY87 cost guidelines = "Galileo class fault detection & correction".	205
Fault Protection shall be implemented on the spacecraft only where required action cannot be successfully or effectively accomplished by ground intervention.	207
With the exception of project approved single point failures as specified in 699-070, no single point failure of any electrical, mechanical or electro mechanical piece-part shall prevent: a) successful target rendezvous, orbit insertion, or penetrator or probe release, b) data acquisition from more than one instrument, or c) acquisition of engineering data not directly related to the failure.	206
4 .TEST AND OPERATIONS REQUIREMENTS	
4.1 ..LAUNCH OPERATIONS	
No power shall be required from the launch vehicle.	60
The spacecraft shall be assembled at SAF and prepared for launch before leaving JPL per Galileo ship and shoot philosophy (modular dissassembly at Pasadena and reassembly at the Cape for launch).	5886
KSC operations shall be limited to launch readiness activities, launch vehicle interface verification, and launch countdown tests. Rationale: This concept is known as 'ship and shoot' and is intended to minimize the cost of KSC operations by short operations, consistent with safety shipping constraints and launch vehicle integration requirements.	209
The launch vehicle shall provide water lines to cool the RTGs.	2614
The launch vehicle shall provide nitrogen lines for prelaunch purge of the spacecraft science instruments.	2615

Figure 5. Sample Document Page With Few Options Selected

Page No. 1 MISSION REQUIREMENTS ON SPACECRAFT SYSTEM

	Value Goal Capability	ID
**** MMII TRACER KEY PHRASE RQMTS PRINTOUT **** **** List 25 ****		
---- BODY FIXED POINTING REQUIREMENTS ----		
POINT THE HGA TO EARTH	3.1mrad	2736
POINT THE PENETRATOR/LANDER IN INERTIAL SPACE Missions: CRAF	17.5 mrad	4939
POINT PROBE SPIN/EJECT MECHANISM INTERFACE IN INERTIAL SPACE Missions: CASSINI	TBD 12 mrad	5948
MANEUVERS IN ANY ORIENTATION		3209
---- PROPULSIVE MANEUVERS ----		
PROPULSIVE MANEUVERS GREATER THAN 1.0 M/S PROPORTIONAL MAGNITUDE ERROR	0.34%	2520

Figure 6. Sample Viewgraph

Figure 7 is a "Selected List" demonstrating the flow down of one branch of requirements from level 1 through level 5.

IMPLEMENTATION

Hardware Description

TRACER is currently developed on an IBM PC AT with a 30 meg hard disk. It could be used on any PC-compatible, MS-DOS machine with a hard disk. Use without a hard disk would be possible, but extremely cumbersome.

The executable files that make up TRACER require approximately 300k of RAM. During execution, additional RAM is required for data, indices, memory variables, and external programs called by TRACER; so it is better to have 512k or 640k of RAM.

It is a good idea to have some form of backup hardware. A 1.2 meg floppy, 360k floppy, or tape drive are all sufficient (a typical mature project may have up to 10 megabytes of data, so it may be a bit tedious to use 360k floppies for backups). The backup process could be enhanced significantly by using any of the commercially available file-compacting programs, but none are incorporated into TRACER.

TRACER REQUIREMENTS FLOW DOWN EXAMPLE

Value
Goal
Capability   ID

**** MMII TRACER FULL STATEMENT, NOT KEY PHRASE,
RQMTS PRINTOUT ****
**** LIST 23 ****
— LEVEL 1 (NASA) —

The MMII Project shall use the Titan IV/Centaur launch vehicle.   7369
Child IDs: 051-1 3217+ ,

FUNDAMENTAL REQMT ON PROJECT LEADS TO MASS CONSTRAINT

— LEVEL 2 (PROJECT) —

The wet spacecraft mass shall not exceed 5327 kg.   5327 kg   3217
Rationale: Launch vehicle capability less launch margin and launch adapter.
Parent IDs: +01S 7369 ,
Child IDs: 051-1 3202+ , 051-1 3215 ,
051-1 3218 , 205 0 ,

THIS ONE REQMT PRODUCES THREE RELATED REQMTS WHICH "LINK UP" TO THE PARENT

The mass of the dry spacecraft including all contingencies and reserves shall not exceed: 1600 kg   3218
Rationale: Assigned mass of spacecraft.
Parent IDs: 051-1 3217 ,
Child IDs: 051-1 5046# , 205-3-230 0 ,
500 0 ,
Missions: CRAF

The propulsion module subsystem shall have an Isp as shown in T051-1:2.1.7- 4.   Table   3202
Reference: T051-1:2.1.7- 4
Rationale: Isp for small main engine burns not defined.
Parent IDs: 051-1 2727 , +051-1 3217 ,
Child IDs: +205-3-160 7225 ,

THESE ARE DERIVED REQMTS

The spacecraft shall hold 3450 kg of main engine fuel plus/minus TBD [10 representative] kg.   3450 kg TBD kg   3215
Rationale: The tanks shall be filled to this value. Tolerance defined by precision of measurement needed fuel plus hold up and residual. Assumes engine ISP = 326s, S/C mass = 1600 kg, Delta V = 3456 m/s.
Parent IDs: 051-1 3217 ,
Child IDs: 051-1 3214 , +205-3-120 7285 ,
205-3-160 5384 , 500 0 ,

ONE CHILD IS NOW FOLLOWED

— LEVEL 3 (SPACECRAFT SYSTEM) —

The spacecraft shall hold 3450 kg of main engine fuel plus/minus TBD [10 representative] kg.   3450 kg TBD kg   5384
Pass-parent source: 051-1 3215
Rationale: The tanks shall be filled to this value. Tolerance defined by precision of measurement needed fuel plus hold up and residual. Assumes engine ISP = 326s, S/C mass = 1600 kg, Delta V = 3456 m/s.
Child IDs: 205-4-2010 0 , +205-4-2010 7283 ,

S/C SYSTEM FUEL ALLOCATION IS RECOGNIZED AND SENT TO THE PROP. MODULE SUBSYSTEM

— LEVEL 4 (S/C SUBSYSTEM) —

The pressurant tank shall be designed to hold 10.7 kg (23.6 lbm) of helium at 257 bars (3720 psia) and 45 deg C (113 deg F).   7283
Comment: Bipropellant tanks and hydrazine tank contain 0.2 kg (0.4 lbm) of helium at launch for a total of 10.9 kg (24 lbm) launched.
Parent IDs: 205-3-160 5384+ , 205-3-160 5389+ ,
205-3-210 5937+ ,
Implementation: PMSRD, Para. 3.2.1
Missions: CRAF

ONE EXAMPLE OF THE FUEL ALLOCATION IS THE PRESSURANT TANK DESIGN (THERE ARE OTHERS AS WELL)

THE RESULTING REQMTS ARE IMPLEMENTED AT THE SUBSYSTEM LEVEL IN A REQMTS DOCUMENT IN A SPECIFICATION OR CONTRACT IN THIS EXAMPLE

Figure 7. Requirements Flow Down Example

A laser printer for producing hard copy is also valuable.

Software Description

Operating System

PC-DOS version 3.1 is the operating system used during development and operation of TRACER. Earlier versions should work as well, since TRACER only uses features found in earlier versions.

Commercial/External Subroutine Usage

TRACER is compiled using the Summer 1987 release of Clipper, which is a dBASEIII compiler produced by the Nantucket Corporation.

The .OBJ files produced by Clipper are linked using the PLINK86-Plus linker produced by Phoenix Technologies, Ltd. and provided by the Nantucket Corporation for use with the Clipper compiler.

The Norton Editor is the text processor used to write the source code for TRACER.

Any external word processor can be invoked from TRACER to edit the memo fields of TRACER databases. The only limit is that the word processor must fit in the memory allocated for external programs.

System Build

The following is a description of the files used to develop TRACER.

The source code is contained in nine files. Object files are created and are later linked to become the executable files. The executable code produced by the Linker consists of four files. Batch files simplify this development.

Design Architecture

The TRACER executable files consist of one main file and three overlays. The main file is loaded when TRACER is invoked. The overlay files are loaded by TRACER one at a time as needed.

Functional Diagrams

Additional disk files contain diagrams and listings to help describe TRACER. Many of these files were created by a program by WallSoft Systems Inc., "The Documentor".

OPERATION

The TRACER is currently operated as a single input system with multiple batch capability for selected non-interactive functions. Input to the system is by paper forms (figure 8), redlined markups of existing reports and documents, or word processor inputs with format similar to the paper input forms. Additional information may be added at any time. Changes are controlled through a procedure built into TRACER and can by further controlled by a formal project change control system.

Reports may be run at any time by special request, this usually requires a few minutes to perhaps an hour or so depending on the document length and complexity of the format chosen. They can be printed using an on-line laser printer.

At the end of each day, the full set of project documents with a standard format is queued up for overnight building from the database as modified that day. The resulting documents are automatically transferred to a VAX and each is thus available, completely up to date, to any user with a computer terminal connected to the local area network. They can be browsed using several search type features or downloaded to the users word processor. A companion file of all requirements in the database allows searches based on arbitrary criteria to find and display any requirement in the entire system almost instantly.

CONCLUSION

TRACER as currently implemented is capable of supporting the requirements documentation and tracing of a full-up project containing more than 25,000 requirements, such as an interplanetary spacecraft project. It is demonstrating its usefulness currently on the Mariner Mark II CRAF and Cassini Projects. It provides convenience and improved visibility and control.

As funding becomes available, the TRACER will be ported to a VAX, and using ORACLE, it will be made fully multiple-user and interactive. Detailed design requirements for the interactive version have not been fully worked out at this time.

ACKNOWLEDGEMENTS

P. Richard Turner provided funding, scheduling, promotion, and overall management for TRACER. Richard L. Stoller led the development of functional requirements for the system, integrated project experience into the design concepts, and provided technical direction and critique throughout the design and operations development process. E. Ted Neville created the functional design, the detailed design, and performed all the programming. Karen A. Boyle provided operational insight and efficiency to the design as database manager. Barrie Gauthier, David Henderson, Len Jaffe, Sandy Krasner, Sima Lisman, Anne Matheson, Paul Sutton, Grace Tan, and Carl Wertz provided helpful suggestions gained through application to a project. Ronald F. Draper committed to the use of TRACER for the Mariner Mark II CRAF and Cassini project. For this paper, Jackie Akers typed the manuscript, and Marysha Cleary and Jeanne Collins edited the text. Thanks are expressed to all.

REFERENCES

1. G. Giffin, J. Skinner, and R. Stoller, "The JPL Functional Requirements Tool", AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, Jan. 12-15, 1987.

2. Dorfman, M. and Flynn, R., "ARTS - An Automate Requirements Traceability System," Journal of Systems and Software, Vol. 4, No. 1, April, 1984, pp 63-74.

3. Pirnia, S. and Hayek, M. J., "Requirements Definition Approach for an Automated Requirements Traceability Tool," Proceedings of the IEEE 1981 National Aerospace

TRACER INPUT FORM
MMII CRAF & Cassini

New Reqmt/Header____
Date this form prepared:___/___/___Originator________________ Update____
Delete____

ALL PARENTS, PEERS, AND CHILDREN WILL BE MADE "BINDING" UNLESS THE DOCUMENT NUMBER IS PRECEDED BY "" TO INDICATE "PENDING"

IDENTIFY PARENT REQUIREMENT(S): [Project number]
_______________:_____,_______________:_____,_______________:_____,
_______________:_____,_______________:_____,_______________:_____,
_______________:_____,_______________:_____,_______________:_____,

THIS REQUIREMENT Document number______________:ID #__________
Identify Peers__________:ID #____
__________:ID #____ Duplicate parent___ Pass parent____
__________:ID #____ Key Word__Inform__

Paragraph No. [of Heading] __________________________
Statement-(Is this a Heading?___Reqmt?___Descr?___) [delimit 'key words']

Reqmt Value:_______________Goal:_______________Capability:______________
Reference:____
Meet parent reqmt? Meet(M)___No(N)___NotSure(?)___AcceptCap(C)___OK(O)__
Comment:

Firmness: Firm(F)___,Soft(S)___,Reviewing(R)___,
Clarifying(C),Contradicting(X)

Verification [enter one or more "methods" for desired configurations]
(ANALYSIS (A), INSPECTION (I), DEMO (O), TEST (T))
CONFIG:

SYS(S)	S/S(B)	ASSY(A)	PART(P)	STV(T)	TV(V)	VIB(N)	AC(C)	SHOCK(K)	OTHER(O)

APPLICABLE MISSIONS (1)___(2)___(3)___(4)___(5)___(6)___(7)___(DRIVER)___

SELECT CHILD(REN):
_______________:_____,_______________:_____,_______________:_____,
_______________:_____,_______________:_____,_______________:_____,
_______________:_____,_______________:_____,_______________:_____,
_______________:_____,_______________:_____,_______________:_____,
_______________:_____,_______________:_____,_______________:_____,

Figure 8. Sample Requirement Input Form

<u>and Electronics Conference</u>, Vol. 1, Dayton, Ohio, May 19-21, 1981, E.I. Conference No. 01320, available from IEEE Service Center (Cat. n81CH1670-9), Piscataway, NJ, pp 389-394.

4. Sciortino, J. and Dunning, D., "Software Traceability, Requirements Testability, and Auditing Model," <u>Proceedings of the AIAA/IEEE 6th Digital Avionics Systems Conference</u>, Baltimore MD, Dec. 3-6, 1984, E. I. Conf. No. 06063, AIAA (CP 8411), New York, NY, pp 159-166.

5. LaGrone, D. and Wallach, E, "Requirements Traceability using DSSD," <u>Tooling Up for the Software Factory, Feedback '86: DSSD Users Conference 13/1-18</u>, Overland Park, KS, Oct 7-9. 1986, publ. by Ken Orr & Associates, Topeka, KS.

6. Yourden, E. and Constantine, L., <u>Structured Design</u>, Yourden Press, 1978.

7. Weiner, M. and Chi, F., "SADT and TMTS - A Marriage of Complementary Management Concepts ... Structured Analysis and Design technique; Technical Management Tracking System," <u>Proceedings of the International Automatic Testing Conference</u>, San Francisco, CA, Nov 3-5, 1987, (A88-36528 14-59) New York, IEEE, Inc., 1987, pp 323-327.