DEVELOPMENT OF A CHANGE MANAGEMENT SYSTEM

Cathy Bonifas Parks
Sterling Software, Inc.
Ames Research Center
Moffett Field, California

INTRODUCTION

The complexity and inter-dependence of software on a computer system can create a situation where a solution to one problem causes failures in dependent software. In the computer industry, software problems arise and are often solved with “quick and dirty” solutions. But in implementing these solutions, documentation about the solution or user notification of changes is often overlooked, and new problems are frequently introduced because of insufficient review or testing. These problems increase when numerous heterogeneous systems are involved. Because of this situation, a change management system plays an integral part in the maintenance of any multi-system computing environment. At the NASA Ames Advanced Computational Facility (ACF), the Online Change Management System (OCMS) was designed and developed to manage the changes being applied to its multi-vendor computing environment. This paper documents the research, design, and modifications that went into the development of this change management system (CMS).

RESEARCH INTO CHANGE MANAGEMENT SYSTEMS

Change management systems have been defined in as many ways as they have been named. Numerous articles document locally-developed and vendor-supplied software management packages and change management systems. Research into the requirements of a CMS revealed the following definitions.

An article describing “change control software” stated, “the primary purpose of a change control system is to impose management control over the application development environment without affecting productivity” (ref.1). The author proposed that a complete change control system should provide MIS organizations with source code change management, audit trails of the changes, synchronization between load modules and source, task and project reporting, management control over parallel development, and options for handling control and quality assurance groups (ref. 2).

Another article described “change management” as a means of ensuring accountability, stating, “It is the function of change management to control the implementation of the new system—to schedule the moment on which it is installed to ensure that all affected users are aware of the change and to control backout procedures” (ref. 3).

A more encompassing view, incorporating control of the complete configuration, was described as “configuration control.” The author stated, “it should be possible to track source and object code, load modules, test data, documentation, and so forth automatically, and to define all applications components" (ref. 4).

But, regardless of what the definition for change management is, or to what extent one feels it covers the total system configuration, there are basic components common to most change management systems. Each of these components plays an important role in the success of the CMS. The general components that are identified in the numerous articles on change management include:

- Software maintenance packages for developing, maintaining, and controlling code.
- Management forms and reports to record change requests and problems.
- Change review processes to reduce the number of "quick fixes" and to promote planning of the changes.
- Documentation and communication of the changes.
- Change implementation and backout procedures.

When selecting a change management system, it is imperative that the chosen package meets the needs of your environment; but the single most important feature to have in your system is "ease of use." The users must be willing to use the system in order for any change management system to be successful.

CHANGE MANAGEMENT SYSTEM REQUIREMENTS AT ACF

At the Advanced Computational Facility at NASA Ames Research Center, the need for a change management system was recognized, but because of the number of heterogeneous systems that were being maintained, and the small amount of software development that was being done, our needs appeared unique when comparing them to the needs addressed by most systems. The majority of changes that needed to be managed at the ACF were modifications to the site-controlled parameters and the configurations of two Cray supercomputers and connected Digital, Silicon Graphics, and Sun Systems. Prior to the implementation of a CMS, changes to these systems often affected our staff and users, but notifications about the changes were not being made. We also experienced problems in which changes to one area of a system affected other areas, but because of insufficient review of the changes, the effects were not noticed until the changes were implemented.

A recent article best defines the CMS requirements that the ACF had when the author described the "software configuration management" function as filling the "need for some organization to ensure that all parties know how to request a change, that a change is necessary, that all affected parties agree with the change, that all parties are informed of the impending change, and that there is a record of all changes made, who made them, when they were made, and why they were made" (ref. 5).

Similar to this definition of the software configuration management function, our primary requirement for a change management system was to provide notification of changes to the ACF staff and the system users. Other important requirements for this system included the need for review of the changes, completion of documentation about the change, and tracking the known problems identified in the various systems. Because of the small amount of software development, there was no need for a software maintenance package within our CMS.

DESIGNING A CHANGE MANAGEMENT SYSTEM

The first change management system for ACF was defined in July 1989. The stated goals of this system were to ensure that all parties were informed of changes and that the impacts of the changes were considered before changes were made. This CMS consisted of seven forms that were used to manage the system, including three forms to document the problem and solution surrounding the change, track the review of the change, and describe the impacts of the change and schedule its installation. Also included were three logs to provide management summaries of the open logs, approval status of the logs, and completed system changes. The final form was the change notification that was sent via e-mail to ACF staff members.

The Systems Group within ACF used this system for two months before it was expanded to include the rest of the staff. In August 1989, a survey was sent to the staff to determine whether the expectations for the CMS were being addressed. The responses received in this survey indicated that the system was too complicated, stating that the best way to ensure continued usage of this system was to simplify the design. As a result, the components of the CMS were reduced to include only three forms: the tracking form (figs. 1-3) which contained the problem, solution, and impacts of the change, as well as the review committee's approval status; the problem log (figs. 4-5) which was used to report the status of all logs on a weekly basis; and the e-mail notification that informed the staff of changes.

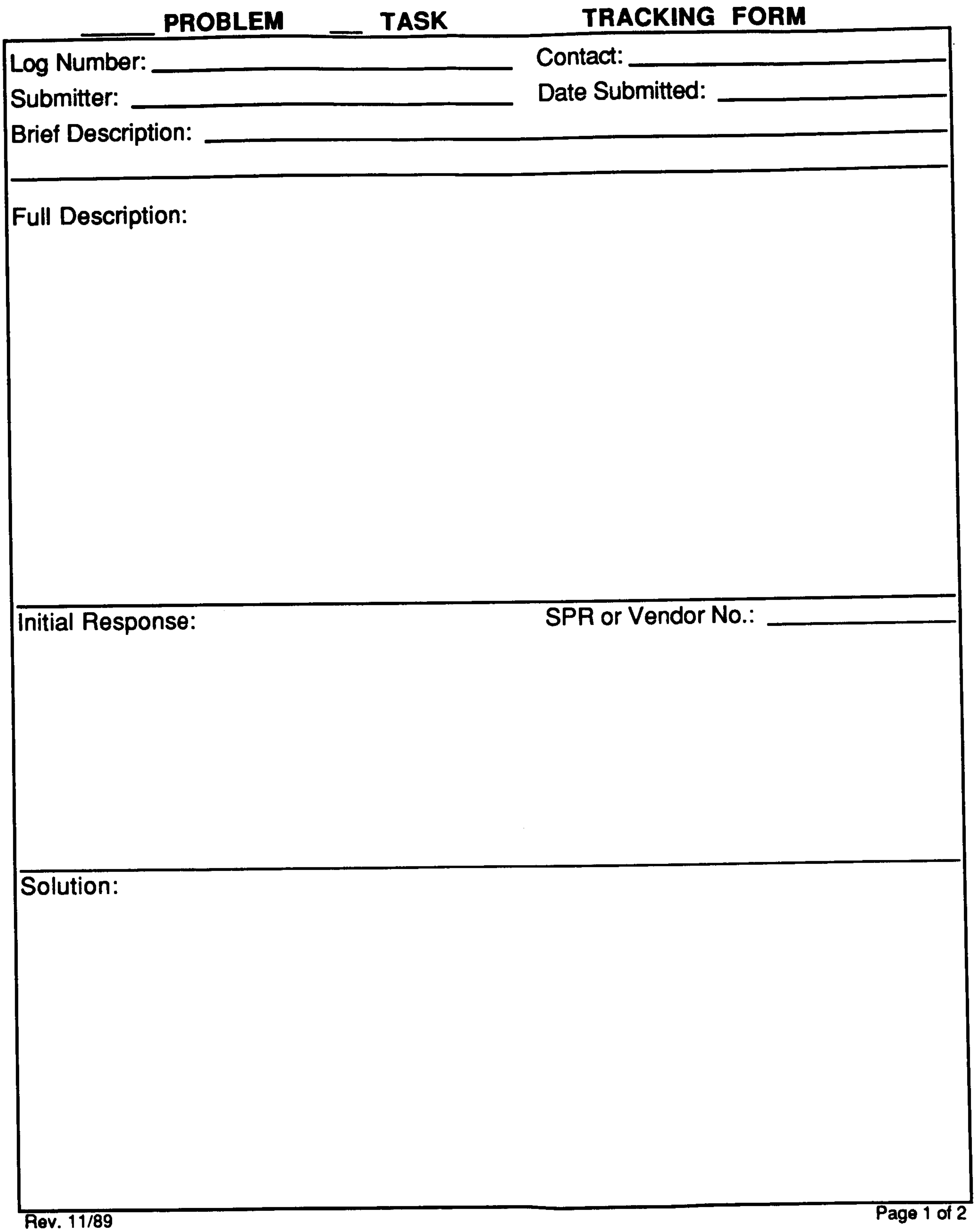
The survey also asked what features were required in the CMS. The majority of the responses indicated that the required features included the documentation of known problems, internal notification of changes, and the review of the changes. The components of the CMS handled these features.

The revised change management system was used within the ACF from August 1989 through May 1990. The procedure for using this system was as follows:

1. A problem or change request was reported by filling out a tracking form that included the problem description and any workarounds for the problem. This form was given a log number and assigned to an analyst.

2. When a solution was available, the solution portion of the tracking form was completed and the form was submitted to the review committee.

3. The review committee approved implementation of the change, or requested that additional work and/or notification be done before the change was implemented.



_____ PROBLEM __ TASK TRACKING FORM

Log Number: ____________________ Contact: ____________

Submitter: ____________________ Date Submitted: ____________

Brief Description: ________________________________

____

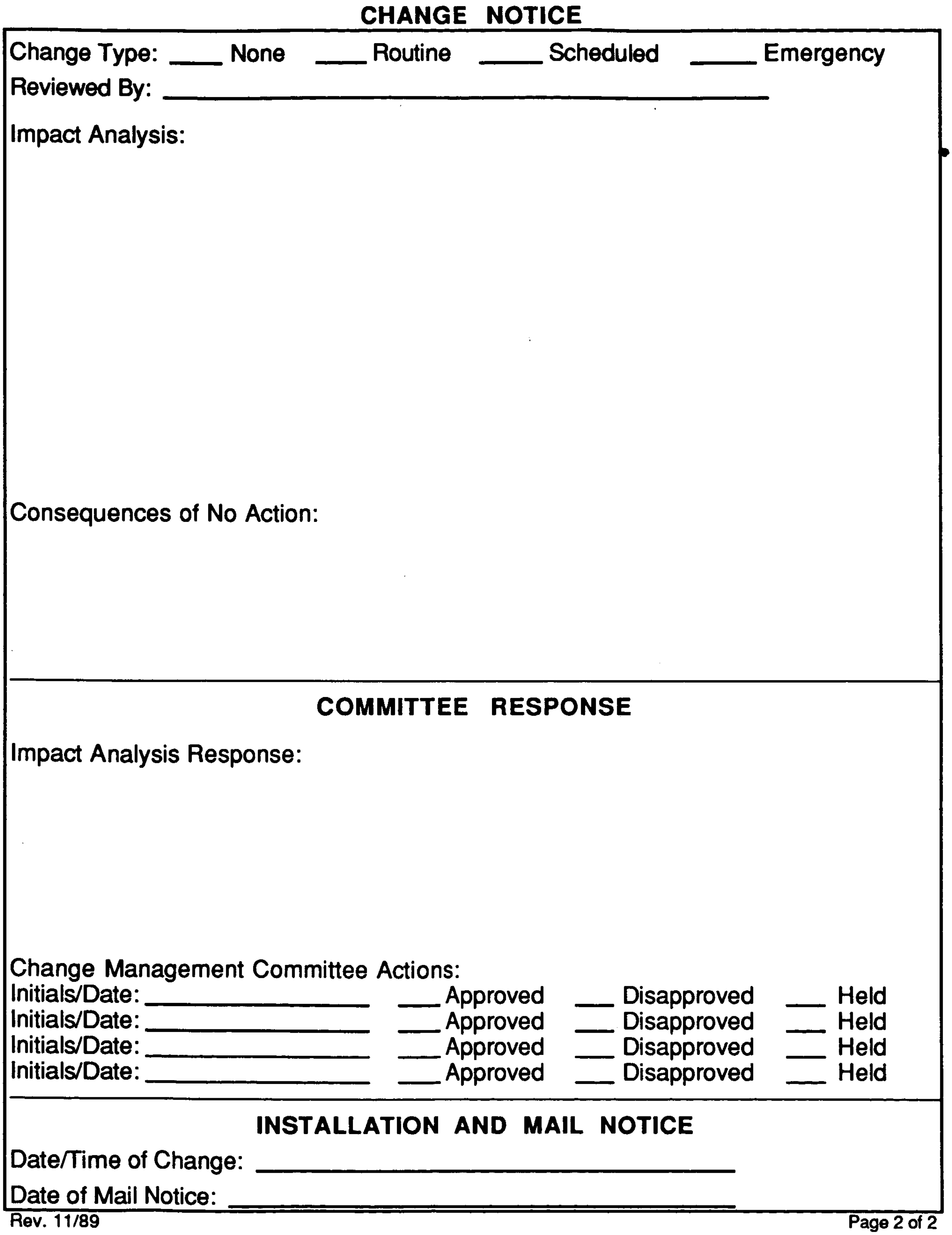
Full Description:

Initial Response: SPR or Vendor No.: ____________

Solution:

Rev. 11/89 Page 1 of 2

Figure 1. Tracking form (side 1).



CHANGE NOTICE

Change Type: ____ None ____ Routine _____ Scheduled _____ Emergency

Reviewed By: ____

Impact Analysis:

Consequences of No Action:

COMMITTEE RESPONSE

Impact Analysis Response:

Change Management Committee Actions:

Initials/Date: ____________________ ___ Approved ___ Disapproved ___ Held

Initials/Date: ____________________ ___ Approved ___ Disapproved ___ Held

Initials/Date: ____________________ ___ Approved ___ Disapproved ___ Held

Initials/Date: ____________________ ___ Approved ___ Disapproved ___ Held

INSTALLATION AND MAIL NOTICE

Date/Time of Change: ______________________________________

Date of Mail Notice: _______________________________________

Rev. 11/89

Page 2 of 2

Figure 2. Tracking form (side 2).

Tracking Form

Definition of Fields:

Problem/Task: Identifies whether the log documents a problem or a task.

Log Number: Identifying number for the problem log.

Contact: Analyst assigned to resolve the documented problem.

Submitter: Person submitting the problem log.

Date Submitted: Date the log was submitted.

Brief Description: One line description of the problem.

Full Description: Detailed description of the problem.

Initial Response: Description of applied workaround or initial suggestion for a resolution for the problem.

SPR or Vendor No: Assigned vendor number if this is a vendor's problem.

Solution: Description of the problem solution.

Change Type: Type of change installed.

None	no change occurred.
Routine	pre-approved common change.
Scheduled	change which has been scheduled.
Emergency	critical change, pre-approval not required.

Reviewed By: Analyst who reviewed the problem solution.

Impact Analysis: Impacts the change will have on users and staff.

Consequences of No Action:
Consequences of not making the change.

Impact Analysis Response:
Further actions requested by the Review Committee.

Change Management Committee Actions:
Initials, date, and approval indication from the committee members.

Date/Time of Change: Date and time the change was installed.

Date of Mail Notice: Date the staff notification was sent.

Figure 3. Tracking form field descriptions.

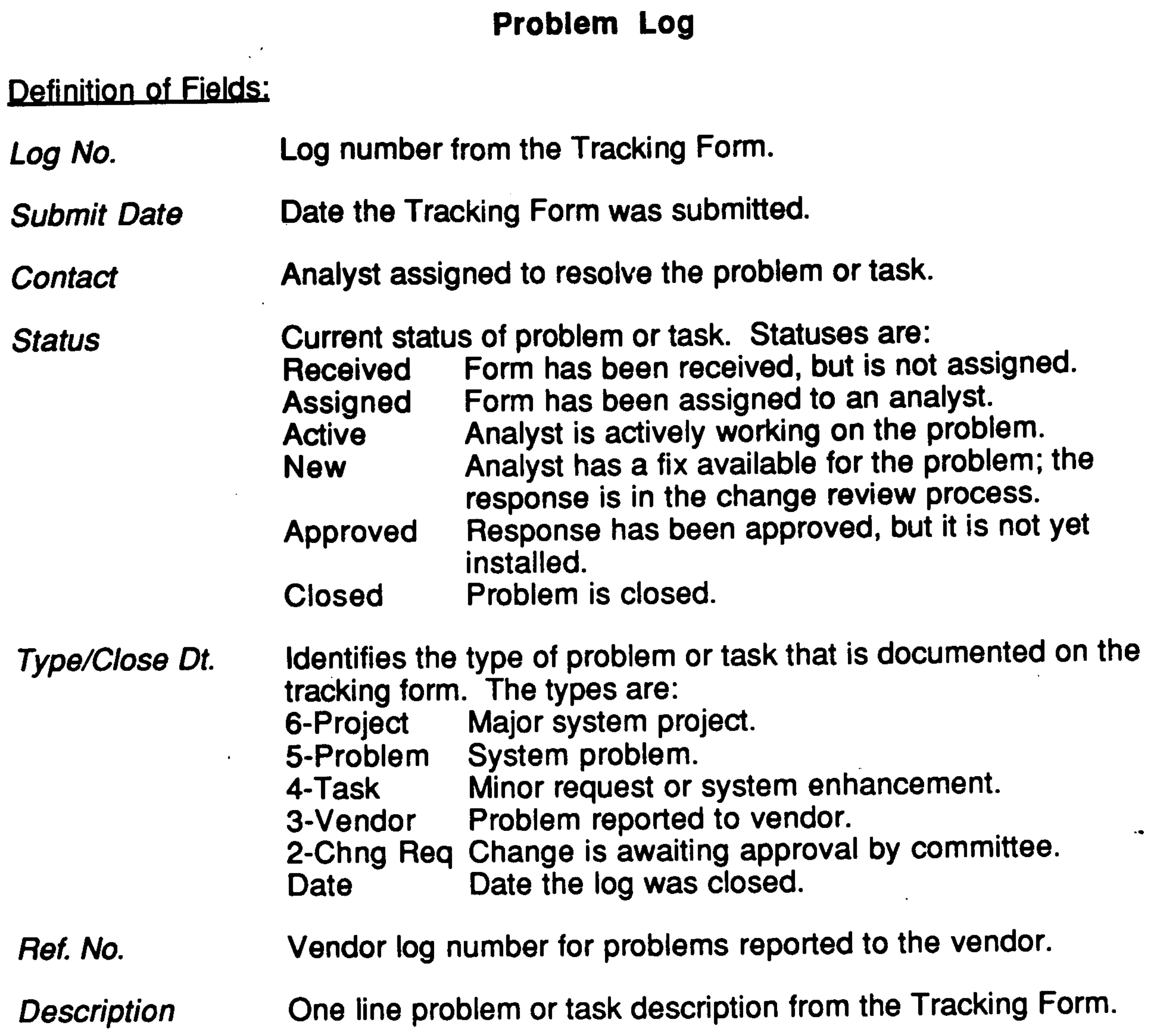
PROBLEM LOG

Log No.	Submit Date	Contact	Status	Type/ Close Dt	Ref. No.	DESCRIPTION

(Date/Time of Report)

Page 1

Figure 4. Problem log.



Problem Log

Definition of Fields:

Log No. Log number from the Tracking Form.

Submit Date Date the Tracking Form was submitted.

Contact Analyst assigned to resolve the problem or task.

Status Current status of problem or task. Statuses are:

Received	Form has been received, but is not assigned.
Assigned	Form has been assigned to an analyst.
Active	Analyst is actively working on the problem.
New	Analyst has a fix available for the problem; the response is in the change review process.
Approved	Response has been approved, but it is not yet installed.
Closed	Problem is closed.

Type/Close Dt. Identifies the type of problem or task that is documented on the tracking form. The types are:

6-Project	Major system project.
5-Problem	System problem.
4-Task	Minor request or system enhancement.
3-Vendor	Problem reported to vendor.
2-Chng Req	Change is awaiting approval by committee.
Date	Date the log was closed.

Ref. No. Vendor log number for problems reported to the vendor.

Description One line problem or task description from the Tracking Form.

Figure 5. Problem log field descriptions.

4. Upon approval of the change, an implementation date was set for the installation of the change.

5. After installation of the change, an e-mail notice was sent to the ACF staff to inform them of the change.

After the initial entry of problems into the CMS, the usage of this system fluctuated between 13 and 36 logs entered each month, with an average of 27 new logs per month. The peak months of usage coincided with memos to the staff emphasizing the importance of this system. The review process for the logs also fluctuated greatly. The committee initially met once a week to review proposed changes, but eventually each member individually reviewed the logs. Because of this process, review of a log could take from one day to one month, and numerous changes were implemented without review.

In May 1990, the CMS underwent a review of its effectiveness when a second survey was sent asking the staff to evaluate the CMS. The areas of "documentation of known problems" and "weekly reporting of CMS information" received the highest scores for the CMS features. Both of these categories received an average score of 7.0 on a scale of 1 (poor) to 10 (great). Internal notification of changes received a 6.6 score, but the category for "change review" received only a 5.3. The overall effectiveness of the CMS, when compared to the previous informal method of managing changes, was rated as 6.7, indicating that the CMS was superior to the old method. But the overwhelming endorsement of CMS was indicated by 100% of the users stating that this system should be continued.

Statistics obtained from the CMS logs showed that 337 problems and tasks were documented during the ten-month period. Of these, 298 logs were closed, with only 42% of them reviewed by the review committee prior to implementation. Of the reviewed logs, 11% had requests for additional action before implementation. Internal notices were issued for 43% of the closed logs.

Recommendations from the survey revealed that 43% of the respondents wanted online database features to ease the use of the system and enhance its capabilities, and 36% mentioned that the review process needed improvement.

DESIGNING AN ONLINE CHANGE MANAGEMENT SYSTEM

In December 1990, a design for an online CMS was completed, and in late April 1991, the paper CMS system was replaced by the Online Change Management System (OCMS). A total of 674 logs had been entered through the old CMS. All open logs were transferred to the new system.

The location for OCMS was specifically chosen to provide easy access for the users. The goals of OCMS included: ease-of-use; the capabilities to enter, transfer, update, comment, close, reopen, and display problem logs; the automatic notification to the OCMS users about new comments and log transfers via e-mail; a command to list recently changed logs; and an audit trail about entries to the logs. The review committee was no longer in effect; instead, each user had the responsibility of staying up-to-date on the information documented in the logs, and they had the capability of adding requests or information to any of the open logs. To accommodate both novice and experienced users, the OCMS provided command and menu-driven modes of operation. An administrator was assigned to maintain the system and to add and delete users of the system.

From May 1991 until mid-November 1991, 280 logs were entered on OCMS, averaging 40 logs per month. This represents a 48% increase over the previous CMS. Of these, 92 logs (33%) received additional comments, and 188 logs were closed.

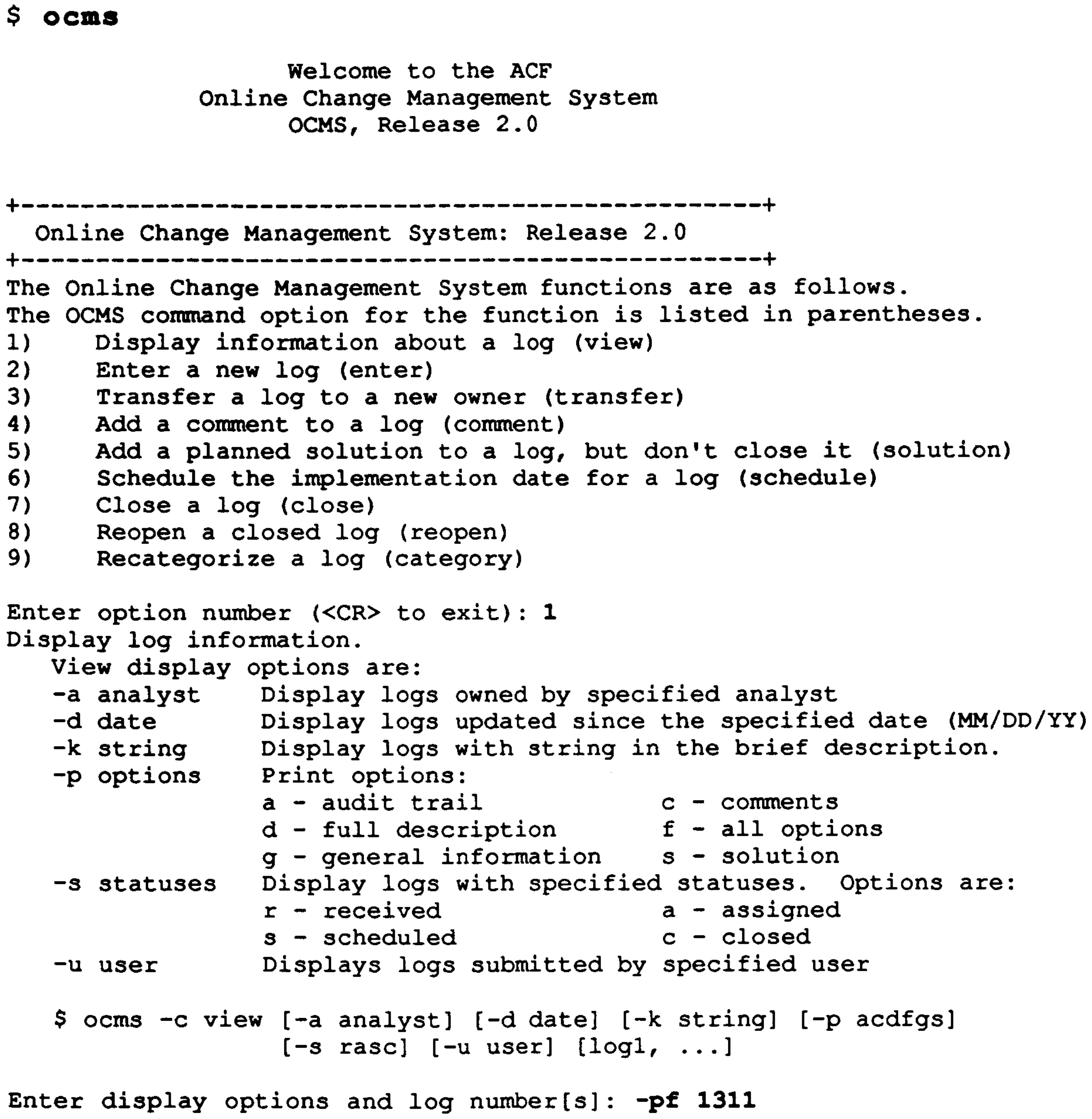
The new OCMS is more widely accepted than the old system, and it has proven to be much easier to use. The process for entering a log (fig. 6) is to enter OCMS and respond to the prompts for the required information. A summary of the new log is displayed when all the information has been gathered. Figure 6 shows an example of the command mode of OCMS; figures 7 and 8 show an example of displaying a full log via the menu mode.

As with the previous CMS, we again reviewed how well we had addressed the goals for OCMS. Enhancement requests were solicited from the staff, and a second version of the OCMS was designed and released in November 1991.

OCMS Version 2 provided categories for the problem or task being documented, which allows for summarizing the types of problems we encounter. It also provided a capability to search for a string in the brief problem description. For the advanced users of this system, the command formats were enhanced to accept unique strings rather than full words as parameters. Further work is in progress to improve the editing capabilities on this system.

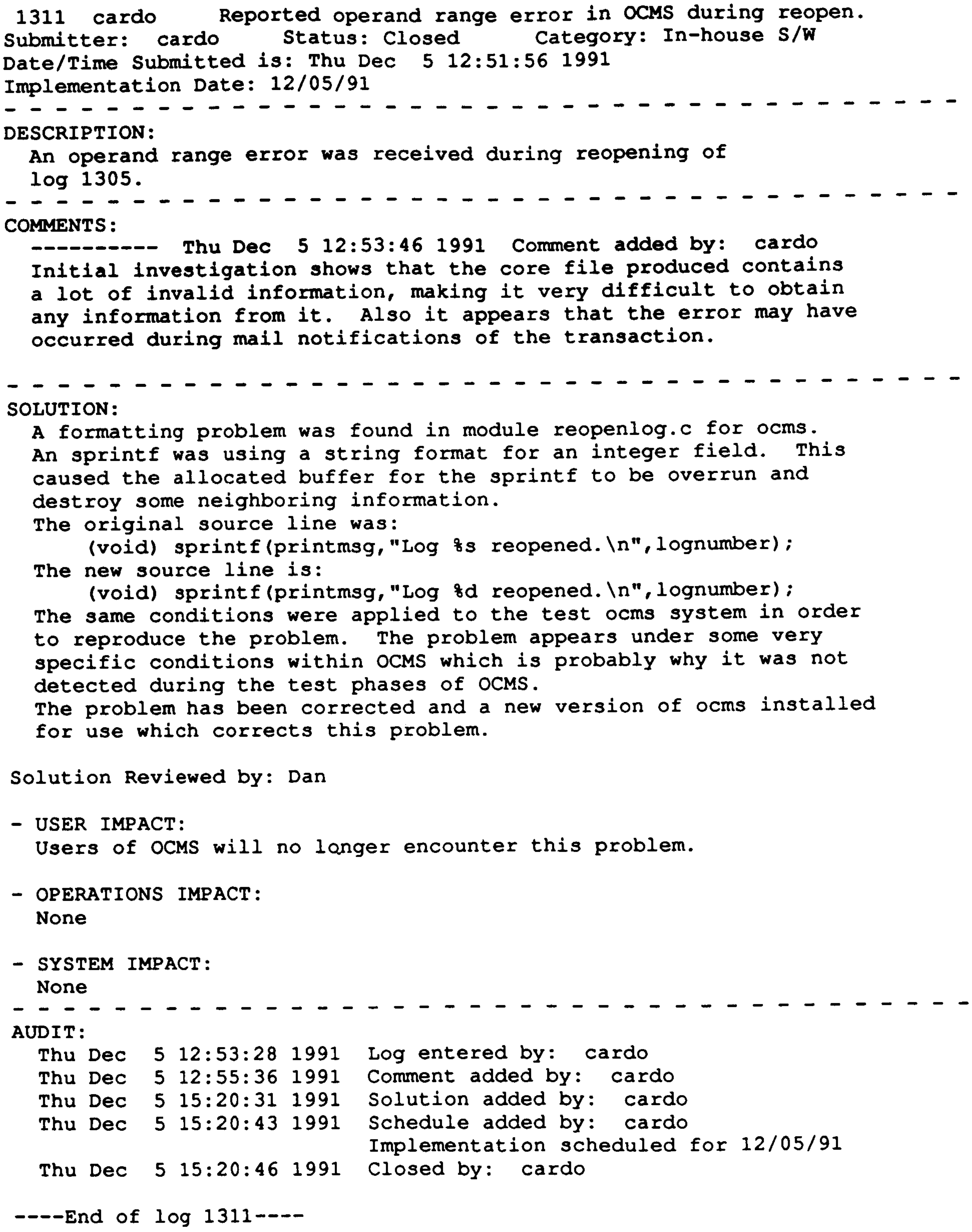
```
$ ocms -c enter

Starting new log.
Enter Log Category
   Valid Categories are:
      1)  Accounting     12) Libraries       22) Sys Problems
      2)  Backups        13) Mailer          23) Sys S/W Maint
      3)  Benchmarks     14) Networks        24) Tapes
      4)  CCF            15) NQS             25) TMX
      5)  Compilers      16) Ops Procedure   26) Training
      6)  Configuration  17) Other           27) ULTRIX
      7)  DMF            18) OWS             28) User Services
      8)  Documentation  19) Performance     29) Utilities
      9)  Hardware       20) Security        30) VMS
      10) In-house S/W   21) Sys Admin       31) Workstations
      11) IOS Problems

Enter Category Number: 1

Submitter:  bonifas   Status: Received    Category: Accounting
Date/Time Submitted is: Fri Dec  6 17:31:54 1991
Implementation Date:  none
Enter brief description:
<Brief problem description is entered here.       >
Enter full description or ctrl-f to include a file (terminate with a '.'):
<Full details of a problem are entered here.                          >
<.                                                                    >

Do you wish to (Commit/Abort/Edit) [Commit]: C
Should this be transferred to you (y/n):  n
Mailing to:  brosen
   99  (None)    Brief problem description is entered here.
Submitter:  bonifas   Status: Received    Category: Accounting
Date/Time Submitted is: Mon Nov  4 13:34:06 1991
Implementation Date: (none)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
DESCRIPTION:

  Full details of a problem are entered here.
```

Figure 6. Log entry on OCMS.



```
$ ocms

                         Welcome to the ACF
                 Online Change Management System
                         OCMS, Release 2.0


+---------------------------------------------------+
  Online Change Management System: Release 2.0
+---------------------------------------------------+
The Online Change Management System functions are as follows.
The OCMS command option for the function is listed in parentheses.
1)     Display information about a log (view)
2)     Enter a new log (enter)
3)     Transfer a log to a new owner (transfer)
4)     Add a comment to a log (comment)
5)     Add a planned solution to a log, but don't close it (solution)
6)     Schedule the implementation date for a log (schedule)
7)     Close a log (close)
8)     Reopen a closed log (reopen)
9)     Recategorize a log (category)

Enter option number (<CR> to exit): 1
Display log information.
   View display options are:
   -a analyst    Display logs owned by specified analyst
   -d date       Display logs updated since the specified date (MM/DD/YY)
   -k string     Display logs with string in the brief description.
   -p options    Print options:
                 a - audit trail           c - comments
                 d - full description      f - all options
                 g - general information   s - solution
   -s statuses   Display logs with specified statuses.  Options are:
                 r - received              a - assigned
                 s - scheduled             c - closed
   -u user       Displays logs submitted by specified user

   $ ocms -c view [-a analyst] [-d date] [-k string] [-p acdfgs]
                  [-s rasc] [-u user] [log1, ...]

Enter display options and log number[s]: -pf 1311
```

Figure 7. Menu display from OCMS.



```
 1311  cardo     Reported operand range error in OCMS during reopen.
Submitter:  cardo     Status: Closed      Category: In-house S/W
Date/Time Submitted is: Thu Dec  5 12:51:56 1991
Implementation Date: 12/05/91
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
DESCRIPTION:
  An operand range error was received during reopening of
  log 1305.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
COMMENTS:
  ---------- Thu Dec  5 12:53:46 1991  Comment added by:  cardo
  Initial investigation shows that the core file produced contains
  a lot of invalid information, making it very difficult to obtain
  any information from it.  Also it appears that the error may have
  occurred during mail notifications of the transaction.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
SOLUTION:
  A formatting problem was found in module reopenlog.c for ocms.
  An sprintf was using a string format for an integer field.  This
  caused the allocated buffer for the sprintf to be overrun and
  destroy some neighboring information.
  The original source line was:
      (void) sprintf(printmsg,"Log %s reopened.\n",lognumber);
  The new source line is:
      (void) sprintf(printmsg,"Log %d reopened.\n",lognumber);
  The same conditions were applied to the test ocms system in order
  to reproduce the problem.  The problem appears under some very
  specific conditions within OCMS which is probably why it was not
  detected during the test phases of OCMS.
  The problem has been corrected and a new version of ocms installed
  for use which corrects this problem.

Solution Reviewed by: Dan

- USER IMPACT:
  Users of OCMS will no longer encounter this problem.

- OPERATIONS IMPACT:
  None

- SYSTEM IMPACT:
  None
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
AUDIT:
  Thu Dec  5 12:53:28 1991  Log entered by:  cardo
  Thu Dec  5 12:55:36 1991  Comment added by:  cardo
  Thu Dec  5 15:20:31 1991  Solution added by:  cardo
  Thu Dec  5 15:20:43 1991  Schedule added by:  cardo
                            Implementation scheduled for 12/05/91
  Thu Dec  5 15:20:46 1991  Closed by:  cardo

 ----End of log 1311----
```

Figure 8. Full view of an OCMS log.

SUMMARY

As the usefulness of our change management system grows, we are also encountering new ways to utilize its features. Tracking tasks through the OCMS has helped in managing analysts' time. The information stored in the logs has been helpful in resolving new problems. And User Services continually use this information to keep abreast of the system changes that may affect the users.

The OCMS users continue to request new features. New requests include a host name, machine type, and level of criticality for each problem, multiple categories for a problem, and automatic notification to the problem submitter of comments added to a log. Management reports have also been requested from this system, in particular, reports that summarize categories of problems and the time it takes to resolve problems.

But, although we have provided a mechanism which addresses our primary goals (documentation of problems, review of changes, and notification of changes), the most difficult requirement for attaining these goals is ensuring that all users utilize this tool. Too often the documentation aspects of OCMS are avoided. As one author states,

> Moans and groans come as soon as that word is mentioned. That dreaded pain acts up in the lower posterior; the activity that is so easily put off until we have absolutely nothing else to do; the most boring part of a technical person's life—documentation. But it is one of the more important aspects of Change Management (ref. 6).

And similarly, providing an inviting, informative, and easy to use change management system is one of the most important challenges for a change management designer.

REFERENCES

1. Di Marcantonio, Joseph: Change Control Software Provides Comprehensive Methodology for Managing the Program Update Cycle. Technical Support, September 1988, p. 33.

2. Ibid., p. 35.

3. Hundley, Kim and Tootill, David: Change Management: Past, Present and Future. Technical Support, September 1988, p. 22.

4. Jander, Mary: Buyers' Guide/Change Control Software: Change Control Meets CASE. Computer Decisions, October 1988, p. 81.

5. Bersoff, Edward H. and Davis, Alan M.: Impacts of Life Cycle Models on Software Configuration Management. Communications of the ACM, August 1991, p. 105.

6. Krukowski, Ray: Building Change Control in the Network. Technical Support, December 1988, p. 42.

BIOGRAPHY

Cathy Parks is a systems analyst with Sterling Software, Inc. She has been working the past four years on the Supercomputing Capability Contract at the NASA Ames' Central Computer Facility. One of her projects on this contract was to design and develop a change management system to control the modifications being applied to the numerous systems maintained through this contract. Cathy has a bachelor's degree in Information Systems Management from the University of San Francisco, and has worked in the computer industry for fifteen years. Before coming to Sterling, she worked for Control Data Corporation and was involved in developing their world-wide call logging system and providing operating system support.