Network Working Group
Request for Comments: 1297

D. Johnson
Merit Network, Inc.

January 1992

NOC Internal Integrated Trouble Ticket System Functional Specification Wishlist ("NOC TT REQUIREMENTS")

Status of the Memo

This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

Abstract

Professional quality handling of network problems requires some kind of problem tracking system, herein referred to as a "trouble ticket" system. A basic trouble ticket system acts like a hospital chart, coordinating the work of multiple people who may need to work on the problem.

Once the basic trouble ticket system is in place, however, there are many extensions that can aid Network Operations efficiency. Information in the tickets can be used to produce statistical reports. Operator efficiency and accuracy may be increased by automating trouble ticket entry with information from the network Alert system. The Alert system may be used to monitor trouble ticket progress. Trouble tickets may be also used to communicate network health information between NOCs, to telcom vendors, and to other internal sales and engineering audiences.

This document explores competing uses, architectures, and desirable features of integrated internal trouble ticket systems for Network and other Operations Centers.

Introduction

This RFC describes general functions of a Trouble Ticket system that could be designed for Network Operations Centers. The document is being distributed to members of the Internet community in order to stimulate discussions of new production-oriented operator-level application tools for network operations. Hopefully, this will result both in more ideas for improving NOC performance, and in more available tools that incorporate those ideas.

PURPOSES OF A NOC TROUBLE TICKET SYSTEM

A good Network Operations Trouble Ticket System should serve many purposes:

1) SHORT-TERM MEMORY AND COMMUNICATION ("Hospital Chart"). The primary purpose of the trouble ticket system is to act as short-term memory about specific problems for the NOC as a whole. In a multi-operator or multi-shift NOC, calls and problem updates come in without regard to who worked last on a particular problem. Problems extend over shifts, and problems may be addressed by several different operators on the same shift. The trouble ticket (like a hospital chart) provides a complete history of the problem, so that any operator can come up to speed on a problem and take the next appropriate step without having to consult with other operators who are working on something else, or have gone home, or are on vacation. In single-room NOCs, an operator may ask out loud if someone else knows about or is working on a problem, but the system should allow for more formal communication as well.

2) SCHEDULING and WORK ASSIGNMENT. NOCs typically work with many simultaneous problems with different priorities. An on-line trouble ticket system can provide real time (or even constantly displayed and updated) lists of open problems, sorted by priority. This would allow operators to sort their work at the beginning of a shift, and to pick their next task during the shift. It also would allow supervisors and operators to keep track of the current NOC workload, and to call in and assign additional staff as appropriate.

It may be useful to allow current priorities of tickets change according to time of day, or in response to timer alerts.

3) REFERRALS AND DISPATCHING. If the trouble ticket system is thoroughly enough integrated with a mail system, or if the system is used by Network Engineers as well as Network Operators, then some problems can be dispatched simply by placing the appropriate Engineer or Operator name in an "assigned to" field of the trouble ticket.

4) ALARM CLOCK. Typically, most of the time a trouble ticket is open, it is waiting for something to happen. There should almost always be a timer associated with every wait. If a ticket is referred to a phone company, there will be an escalation time before which the phone company is supposed to call back with an update on the problem. For tickets referred to remote site personnel, there may be other more arbitrary timeouts such as

"Monday morning". Tickets referred to local engineers or programmers should also have timeouts ("Check in a couple of days if you don't hear back from me"). A good trouble ticket system will allow a timeout to be set for each ticket. This alarm will generate an alert for that ticket at the appropriate time. Preferably, the system should allow text to be attached to that timer with a shorthand message about what the alert involves ("Remind Site: TT xxx") (The full story can always be found by checking the trouble ticket). These alerts should feed into the NOC's standard alert system.

The Alarm Clock can also assist (or enforce!) administrative escalation. An escalation timer could automatically be set based on the type of network, severity of the problem, and the time the outage occurred.

5) OVERSIGHT BY ENGINEERS AND CUSTOMER/SITE REPRESENTATIVES. NOCs frequently operate more than one network, or at least have people (engineers, customer representatives, etc) who are responsible for subsets of the total network. For these individual representatives, summaries of trouble tickets can be filtered by network or by node, and delivered electronically to the various engineers or site representatives. Each of these reports includes a summary of the previous day's trouble tickets for those sites, a listing of older trouble tickets still open, and a section listing recurrent problems. These reports allow the site reps to keep aware the current outages and trends for their particular sites. The trouble ticket system also allows network access to the the details of individual trouble tickets, so those receiving the general reports can get more detail on any of their problems by referencing the trouble ticket number.

6) STATISTICAL ANALYSIS. The fixed-form fields of trouble tickets allow categorizations of tickets, which are useful for analyzing equipment and NOC performance. These include, Mean Time Between Failure and Mean Time to Repair reports for specific equipment. The fields may also be of use for generating statistical quality control reports, which allow deteriorating equipment to be detected and serviced before it fails completely. Ticket breakdowns by network a NOC costs to be apportioned appropriately, and help in developing staffing and funding models. A good trouble ticket system should make this statistical information in a format suitable for spreadsheets and graphics programs.

7) FILTERING CURRENT ALERTS. It would be possible to use network status information from the trouble ticket system to filter the alerts that are displayed on the alert system. For instance, if node XXX is known to be down because the trouble ticket is

currently open on it, the alert display for that node could automatically be acknowledged. Trouble tickets could potentially contain much further information useful for expert system analysis of current network alert information.

8) ACCOUNTABILITY ("CYA"), FACILITATING CUSTOMER FOLLOW-THROUGH, AND NOC IMAGE). Keeping user-complaint tickets facilities the kind of follow through with end-users that generates happy clients (and good NOC image) for normal trouble-fixing situations. But also, by their nature, NOCs deal with crises; they occasionally find themselves with major outages, and angry users or administrators. The trouble ticket system documents the NOC's (and the rest of the organization's) efforts to solve problems in case of complaints.

FIXED FIELDS, FREE-FORM FIELDS, and TT CONFIGURATION

Information in trouble tickets can be placed in either fixed or freeform fields. Fixed fields have the advantage that they can be used more easily for searches. A series of fixed fields also acts as a template, either encouraging or requiring the operators to fill in certain standard data. Fixed fields can facilitate data verification (e.g., making sure an entered name is in an attached contacts database, or verifying that a phone number consists of ten numeric characters). Fixed fields are also appropriate for data that is automatically entered by the system, such as the operator's login id, the name of the node that was clicked on if the trouble ticket is opened via an alert tool, or names and phone numbers that are automatically entered into the ticket based on other entries (e.g., filling in a contact name and phone based on a machine name).

Unfortunately, fixed fields work best where the problem-debugging environment is uniform, well-understood, and stable; that is, trouble tickets work best when their fields are well tailored to the specific problem at hand. It is easy to set up a large number of fields (or even required fields) that are irrelevant to a given problem; this slows down and confuses the operators. Adding structure and validity checking to a field tends to make the data more consistent and reliable, but it also tends to force the operators into longer procedures like menus to get the get the data accepted by the system. It also forces there to be more maintenance on those verification systems (adding new entries as they become new legal options), and in some ways it reduces the accuracy of the system by forcing operators to choose "canned" or authorized responses that may not always represent the situation accurately. Where statistical operational reports are a primary purpose of the trouble ticket system, several fixed fields may be appropriate. If the primary intent of the system is to keep notes for individual problems and to facilitate

communication between operators, then fixed fields may tend to be a hindrance. One reasonable guideline would be that fixed fields are used ONLY where they are automatically filled in by the larger system, or where the information in that field is explicitly used in a report or standard search procedure.

Because of this close relationship between the structure of the ticket and the problem to be solved, it is very very useful to be able to define different ticket types for different classes of problems. This becomes even more true for those many NOCs whose staff are responsible for other types of operations: mainframe operations, workstation administration, help desk functions, or any of the other real-time response functions. Network operations to justify the expense of an operations center. This kind of operation makes economic sense, and is becoming more prevalent. In these kinds of situations it is vital that the same tools that are used for network operations also be available for the other operations. This means that the trouble ticket configurations need to be modifiable by local staff. Commercial RDBMS forms builder and report generator packages and "fourth-generation languages" offer a good start at this, although it is sometimes difficult to integrate full trouble ticket functionality through these systems.

TROUBLE TICKET STRUCTURE

1) HEADERS. Inevitably, a trouble ticket begins with a number of fixed fields. These generally include:

Time and Date of problem start.
Initials or signon of the operator opening the ticket.
Severity of the problem (possibly separating the "customer severity" and the "NOC priority", since these could be different).
A one-line description of the problem for use in reports.

There can be many other fixed fields for specific purposes. There may also be different kinds of tickets for different problems, where the ticket format differs mainly in fixed fields. These include:

Who reported the problem? (Name, organization, phone, email address)
Machine(s) involved.
Network involved (for multi-network NOCs).
User's machine address.
Destination machine address.
Next Action.
Time and date for alarm on this ticket.
Who should the ticket be dispatched to?
Ticket "owner" (one person designated to be responsible overall).

2) INCIDENT UPDATES. The main body of trouble tickets is usually a series of freeform text fields. Optimally, each of these fields is automatically marked with the time and date of the update, and with the signon of the operator making the update. Since updates are frequently recorded sometime after the problem is fixed, however, it is useful to allow the operators to override the current time stamp with the time the update was actually made. (In some implementations, both times will be kept internally).

The first incident update usually is a description of the problem. Since the exact nature of the problem is usually not known when the ticket is first opened, this description may be complex and imprecise. For problems that are reported by electronic mail, it is useful to be able to paste the original message in the ticket, particularly if it contains cryptic or extensive information (such as a user's traceroute output). At least one such arbitrarily-long freeform field seems necessary to contain this kind of output, although it is better to allow arbitrarily long messages at any stage (e.g., so future complex messages can also be archived in the ticket).

Subsequent update fields may be as simple as "Called site; no answer". Some systems allow these kinds of updates to be coded in fixed fields; most use freeform text.

There should always be an indication of what the next action for this ticket ought to be. Again, this may be implemented as a special fixed field, or by convention of using the last line of text.

Advanced systems may also need a facility to allocate the amount of time a ticket is open between multiple sources. A serious NOC will want to use its trouble ticket system to statistically track its performance on responding to problems. (e.g., Mean Time Between Failure and Mean Time To Repair reports). Frequently, though, repairs are stopped at the customer's request. ("It's not that important a machine and I don't feel like coming in--can you defer it until Monday Morning?"). In these cases the ticket needs to remain open, but there needs to be a notation that the ticket is now in "customer time" rather than "NOC time". The durations of "customer time" need to be excluded from MTBF and MTTR reports. Complicated repairs could move back and forth between "NOC time" "customer time" repeatedly. This probably implies that each Incident Update may have a time and date of status change, and that these status changes can be read and aggregated by by reporting programs.

3) RESOLUTION DATA. Once a problem is resolved, it is useful to summarize the problem for future statistical analysis. The following fields have been found to be useful:

- Time and Date of resulation (for outage duration).
- Durations (can be calculated from time of resolution and incident report "customer/NOC time" stamps).
- Resolution (one line of description of what happened, for reports).
- Key component affected (for MTBF and similar reports).
- Checked By -- a field for supervisors to sign off on ticket review.
- Escalated to -- for reports on how many problems require non-NOC help.
- Temp - a database field that can be used to store temporary "check marks" while making statistical investigations.

USER, TROUBLE, and ENGINEERING Ticket System(s)

The primary level of an Network Operations trouble ticket is the "problem" or "trouble": a single malfunctioning piece of hardware or software that breaks at some time, has various efforts to fix it, and eventually is fixed at some given time.

The primary level of an Network Information Center ticket, however, might well be the "user complaint". A single network failure might well produce a large number of individual user phone calls and hence "user complaint" tickets. A NIC may want to use tickets to track each one of these calls, e.g., to make sure each user is informed and satisfied about the eventual resolution of the single hardware problem.

In addition, NOCs (or Engineering Staffs) may want to track systematic problems. The staff may know, for instance, that a particular router is old and fragile, or that a particular section of their network doesn't have enough redundancy. It may be useful to open an "Engineering Ticket" on these known problems, providing a place to record history and notes about the problem, for use in further engineering or funding discussions.

Even further "Meta" tickets could be described, having to do with such issues as whether the current trouble ticket fields, reports, and operation procedures were sufficient to handle current problems.

It would be very convenient to be able to build all of these systems on the same platform, and to allow each type of ticket to easily reference other types. Multiple "user complaint" tickets, then, might might explicitly point to a single "trouble" ticket. Multiple trouble tickets representing independent failures would then point to a single "engineering" ticket, which described the systematic problem. Multiple engineering tickets could point to a single "meta" ticket, if appropriate.

ASSISTED ENTRY AND DATA VERIFICATION

Data (particularly in fixed fields) is only useful for searching if it is entered in consistent formats. A trouble ticket system needs to help operators fill these fields with the correct format of information. This can be done using assisted entry (menus of acceptable choices), verification routines which check against internal lists or external databases (see next section), or other computer checking.

Some database systems allow a customized "help" screen to be associated with each field, helping new (and experienced) operators by making context-sensitive trouble ticket system documentation available at every field.

Very complicated help or operator-guidance systems can be built out of Expert System technology. This could be as simple as help screens, or help screens with database information inserted (e.g., site contact names and phone numbers). Or it could involve hints to the operator, based on current network conditions. Or it might even ask the operator to run tests and to type in the results. (See EXPERT SYSTEMS, below).

INTEGRATION

To be maximally efficient and useful, a Trouble Ticket system needs to integrate well with most of the rest of the NOC tools. These include:

1) OPERATOR WINDOW ENVIRONMENT. A NOC Operator needs access to many pieces of information simultaneously, and therefore is well served by a good windowing environment. The Trouble Ticket system needs to run within this larger windowing system, so that the operator can debug, consult databases, use Email, field alerts, and keep an eye out for other emergencies while working on a trouble ticket. It is also useful to be able to run two trouble ticket sessions simultaneously, for example, to allow an operator to search for related tickets while he is in the middle of updating another ticket. Cut and Paste between these various screens is mandatory, to allow easy recording of technical details in the trouble tickets.

2) ALERT MONITORING SYSTEM. Trouble tickets are often opened in response to machine alerts; it ought to be easy to open a trouble ticket directly from the alert tool. When a ticket is opened this way, information about the alert and the machine involved ought to be automatically filled into the ticket. (There are various opinions about whether trouble tickets ought to be opened

automatically without operator intervention. This operator's opinion is that an operator acknowledgement should be required, but this point is debated enough that designers of a new system probably should support either option).

The Alarm Clock feature of the trouble ticket system also generates alerts. These alerts ought to feed gracefully into the Alert Monitor system, so that the operators will get all of their alerts from one place.

3) DATABASE CONNECTIONS. A good trouble ticketing system will query NOC databases to automatically fill out trouble ticket fields where possible. This can be used for:

- Filling out Network Operator information (e.g., phone number) based on the NetOp's signon id.
- Filling in contact information based on machine name.
- Filling in circuit numbers based on link description.
- Filling in alarm clock or escalation time fields based on the machine or link name and on time of day.
- Filling in machine serial numbers based on configuration database.

4) MACHINE QUERYABLE INFORMATION. It could also be possible for a trouble ticket system to make standard queries of the network itself when a trouble ticket is opened: e.g., the system could request and store current machine configurations whenever a ticket was opened for that machine. On some systems, hardware serial numbers are obtainable by software query directly to the machine.

5) ELECTRONIC MAIL. Problem notification often comes via electronic mail; it must be possible to easily open a ticket and include the original mail message within the ticket as part of the initial problem description. When extremely technical messages come in from network engineers, it is useful to allow those messages to be included verbatim, rather than forcing less technical network operators to rephrase the messages or to force them into predefined formats. Later update messages should also be easily includable. Possibly: tickets could be opened automatically for mail messages to certain mailboxes. A response system saying "Your request has been received and assigned ticket number ####" might be desirable.

Information within trouble tickets must also be easily available (possibly just via the windowing system) for inclusion in Email messages to engineers and others.

Scheduled (e.g., daily) reports must also be easily generated into the Email system.

6) DISPATCHING AND NOTIFICATION SYSTEMS. An important real-time aspect of Network Operations is notifying users, technical contacts, and administrators of various classes of problems. The rules for who gets notified of what can be very arbitrary and complex, and can involve electronic mail, notices in computer conferences, automatic beeper pages, and synthesized voice announcements. It would be good for a trouble ticket system to provide for automatic (or operator initiated) notification of the appropriate channels for the current ticket (based on network, machine, severity of problem, duration of the problem, escalation guidelines, etc).

Databases associated with the trouble ticket system may also have lists of specific people to contact about outages for particular machines. These "who to inform" lists can facilitate customized notification messages directly from the trouble ticket system.

It may also be possible to dispatch experts directly from the trouble tickets system. IBM's ECCO system allows allows customers to directly dispatch Service Engineers from machine interactions. Many NOCs also use computer hooked to modems to automatically page engineers. This kind of dispatching should be available from within the trouble ticket system (along with an automatic note into the trouble ticket that the engineer has been dispatched).

7) OTHER TROUBLE TICKET SYSTEMS. When the NOC generates a trouble ticket, it often immediately calls up a telco or another Internet NOC, who proceed to open their own ticket. The Internet Engineering Task Force User Connectivity Working Group is also proposing a national trouble ticket tracking system, which would need updating from individual NOC trouble ticket systems. A state-of-the-art trouble ticket system could have provisions for transferring tickets and ticket information in and out of other such systems.

8) NETWORK ACCESS. Some older trouble ticket systems assumed that anyone with a need to access the information would obtain a signon and learn to use that system. The range of people with a need for trouble ticket information is now too great to allow this assumption. A good system now needs to be able to support network query for tickets and summary reports, as well as Email delivery of scheduled reports.

9) SUBROUTINE INTERFACE. To allow for ad-hoc and currently unanticipated needs, the trouble ticket system needs to support a full-function set of subroutine calls. These subroutines will allow construction of further trouble ticket functionality not yet specified.

10) EXPERT SYSTEMS. Network debugging is a very promising area for expert system and artificial intelligence applications. But such an algorithm should require access to the alert monitoring system, configuration and change control systems, to the network itself, and also to the information in the trouble ticket system. A good future system then needs to make this information available (probably via the subroutine interface mentioned above), and to also allow the Network Operators to invoke the artificially intelligent debugging from within a trouble ticket (including its output as part of the ticket dialogue).

11) GRAPHICS/REPORT Capability. Statistical and graphical displays about trouble ticket data need to be compatible with tools used to generate reports, news letters, etc.

OTHER CONSIDERATIONS:

1) INTERACTIVE SPEED. The system must be fast enough to be used interactively. NetOps need to answer questions over the phone in real time; good answers cannot be given if every query takes a couple of minutes. More importantly, the NetOps need the trouble ticket system in order to get information necessary to fix the network. If looking for old or currently-open tickets takes more than a few seconds, it won't be done. If updates take very long to make, then updates won't be recorded, or they will be recorded long after the event (with corresponding loss of accuracy). (Our Operators have asked for a single-line "update this ticket with this message" utility that would let them avoid even retrieving the ticket for simple updates!) Any time spent waiting reduces NetOp productivity and Network reliability.

2) BACKUPS AND RELIABILITY. The trouble ticket system is absolutely crucial to both immediate and long-term operation of the NOC. Good systems could back up all data several times an hour to an auxiliary processor. That processor should be accessible for immediate use in case of failure of the primary system.

3) HISTORY AND ARCHIVING. A trouble ticket system is a constantly-growing database system. Old tickets need to be removed from the system at some interval (a year? several years?) and archived. These archives should also be restorable for long-term history processing.

4) PRIVACY AND SECURITY. The ability to enter, append, and modify tickets should be controlled by id and password. Permissions should be specifiable on a per-field basis. General read access to tickets (or portions of tickets) also needs to be restricted,

or else NetOps will be reluctant to be full and candid in their reporting.

UTILITY

There are quite a few ideas in this "Wishlist". Ultimately, what an Operations Center needs is a totally integrated set of tools which completely model all of its activities, and which integrates cleanly with all backup, peer, and vendor NOCs. It is hard to imagine that this whole system could come out of a shrinkwrapped box, even without the local configuration. But most of these facilities do exist, now, in some system. Hopefully, this document will foster an ongoing discussion of ways in which NOC operator-level tools are used in real operations, and will encourage systems implementors and vendors to bring some of this functionality to the aid of real operations. It might even inspire current Operations Centers to add useful features to their current operations.

Security Considerations

This paper does not pose specific new security issues. The systems described herein would be host database applications, however, or even distributed host database applications. All of the normal security considerations for that kind of system would apply. Multiple classes of user access need to be specified for classes of ticket data. Possible security threats include disclosure of network information, disclosure of confidential material (e.g., circuit numbers or home phone numbers), and denial of service to the Network Operations Center leading to degradation of network service.

Author's Address

Dale S. Johnson
Merit NOC
1075 Beal Avenue
Ann Arbor, MI 48109-2112

Phone: (313) 936-2270

Email: dsj@merit.edu

Discussion/comments may be sent to noc-tt-req@merit.edu. The list is maintained by noc-tt-req-request@merit.edu.