Inter-Office Memorandum

To	Communication Protocols	Date	June 14, 1976
From	Ed Taft	Location	Palo Alto
Subject	Event Report Protocol	Organization	PARC/CSL

XEROX

Filed on: <Pup>EventReport.press

The Event Report Protocol (ERP) is a means by which events occurring in our distributed systems may be collected and recorded for later analysis. The immediate application for this protocol is the collection of hardware error reports from Altos so as to facilitate maintenance, but the protocol is not limited to this function.

The principal objective considered in the design of the protocol is that the act of generating an event report be as simple and straightforward to implement as possible. This objective is achieved at the expense of making the processing of received reports somewhat more difficult than it might otherwise be.

This memo covers three topics: specification of a general Event Report Protocol, specialization of this protocol to deal specifically with Alto hardware errors, and description of the initial implementation of an event report recipient process on Maxc.

The General Protocol

The reporting of an event is accomplished by means of an exchange of Pups between a *generator* process and a *recipient* process. The recipient process resides at a well-advertised port (inter-network address), whereas the generator process may transmit reports via an arbitrary port. The means by which a generator discovers the address of an appropriate recipient is not specified by this protocol.

The generator initiates an event report by transmitting an *Event Report Pup*. Upon successful receipt of an Event Report Pup, the recipient returns an *Event Report Reply Pup* with matching Pup Identifier. The generator process is expected to retransmit the Event Report Pup (with the *same* Pup ID) after a reasonable timeout if no Reply is received, and if feasible to take cognizance of any other information available (such as receipt of an Error Pup).

The generator is expected to choose Pup IDs in a manner that will facilitate correct processing by the recipient. In particular, if duplicate suppression or maintenance of chronological sequence is important, then Pups should be sequence-numbered or time-stamped. The general protocol does not specify the choice of Pup IDs.

Similarly, the interpretation of the Event Report Pup’s contents is strictly a matter of convention between the generator and recipient processes. We expect, for example, that Alto hardware error reports will be directed to a different recipient (i.e., a different port) than, say, Bravo usage statistics, and we therefore do not require that the information be formatted or interpreted in a standard way. (Of course, one will probably want to use as a model the specific protocol to be presented in the next section.)

Copyright Xerox Corporation 1979

It is assumed that an event may be reported in a single Event Report Pup, and therefore that separate Event Report Pups describe independent events. The consequence of this property is that event reports may be generated spontaneously, without any need for the generator process to maintain state information between occurrences of events. The recipient will of course need to perform some processing, both to differentiate between reports arising from multiple sources and to filter out duplicates and maintain correct sequence (where required). However, we anticipate that in most applications, such processing will not be done in real time (immediately upon receipt of reports) but rather will be deferred until the data is reduced or analyzed, at which time the analysis program can perform the necessary processing.

The Pup type assignments for the ERP are:

Event Report	240 (octal)
Event Report Reply	241 (octal)

Alto Error Reports

In this section, we specify the manner in which Alto hardware errors are reported using the ERP.

All Event Report Pups are directed to a server process residing at socket 30 on Maxc (i.e., at inter-network address 3#200#30). Since this server may move and we may eventually design a general protocol for dynamically discovering the addresses of widely-used services, report generator programs should be written in such a way that the recipient address is easy to change.

The Pup ID is set to the sending Alto's calendar clock (the one returned by DayTime) so as to identify the date and time of the report. This convention constrains an Alto to send Event Report Pups no more frequently than once every second (the grain of the calendar clock); otherwise, succeeding reports will appear to the recipient to be duplicates of the first. For Alto hardware errors (which will probably be generated by Swat), this does not seem an unreasonable assumption.

The format of the Pup contents will be specified by Bob Sproull, who is implementing the Alto software responsible for reporting hardware errors.

Maxc Recipient Implementation

The ERP recipients are implemented as part of Pupsrv, the system background program that controls Pup server processes such as Telnet and FTP. Pupsrv performs no logical processing on received event reports, but rather merely appends them to a file for later analysis.

The association between ERP recipient sockets and files is established by a file <System>Pupsrv.Run, which Pupsrv reads when it is started. This file contains one or more entries of the form:

```
ERP socket-number, filename
```

For example,

```
ERP 30, <Cucinitti>AltoErrors.Log
```

For each Event Report Pup received on one of the declared sockets, the event is appended to the highest-numbered existing version of the file associated with that socket (a new file is created if necessary). The file is kept open only while information is being appended, so it is possible for an analysis program to rename the file when it is about to process the data contained within it, thereby forcing a new log file to be created next time an event report is received. To reduce overhead, Pupsrv does not write out event reports immediately upon their receipt, but rather buffers them for up to 30 minutes or until its internal buffers overflow.

The event report log file consists of entries written as a sequence of 8-bit bytes. The format of each entry is:

Length of entry, in bytes (including itself) (2 bytes)
Pup source address (6 bytes)
Pup ID (4 bytes)
Pup contents (however many bytes it contained)

Alto programmers interested in making use of the Maxc ERP server should be aware that its operation is relatively expensive. One should take care not to inundate it by sending too-frequent event reports. For example, an event report generated by every keystroke (or even every Bravo command, for example) would be totally unreasonable, whereas a summary generated by the "Quit" command would be perfectly acceptable. If the events of interest occur very frequently or involve gathering and transferring large volumes of information, it is better to buffer the data locally and later transfer it all at once using FTP.