To Pilot Interest

Date November 23, 1977

From H. C. Lauer

Location Palo Alto

Subject Time Of Day Facilities in Pilot

Organization SDD/SD

XEROX

Filed on: <Lauer>TimeOfDay.memo

This memo describes our proposal for facilities for maintaining the time of day and date within a system element. This proposal is predicated on the availability of a reliable, battery-powered Time-and-Date clock in the hardware (in the form of a watch chip or equivalent), and it is characterized by the separation of the time-of-day functions into simple, uniform facilities provided by Pilot and facilities oriented to the human user provided by some form of common software.

General Strategy

We propose that the clock in each D0 processor maintain the time and date according to Greenwich Mean Time. Pilot will implement an operation by which any client can get the current time (GMT) and a private operation by which special programs can correct the time in the case of clock drift, battery replacement, or repair of malfunction. The time will probably be returned in the unpacked format described below (unless the packed format turns out to be more convenient). Common software routines will be provided to convert this value into values of the local time and date and into formats suitable for application programs. These routines will also implement such features as Daylight Saving Time, which depend upon the locality in which the system is installed.

It is the intention of this proposal that *all values of the time and date stored internally within a system element or transmitted over a network in a form intelligible to OIS software be in Greenwich Mean Time.* In particular, Pilot will use the Greenwich Mean Time whenever it records the time, such as the creation date or last reference date for a file. This convention will eliminate much confusion over the meaning of time-stamps and time values communicated by machines scattered over the globe. It will, to a large extent, keep the customer from having to set the time (and get it wrong), and therefore it will allow applications to depend fairly heavily on having a meaningful, consistent value of the time available on any particular system element. It will also allow us to parameterize our software to accommodate local conventions about the official time without having these considerations penetrate the applications themselves.

Note that although the timing facilities are expected to be very reliable and the watch chip will record the passage of time very accurately, the absolute value of the time will not be particularly accurate. I.e., this proposal does not provide a network-wide or universal source of the time. In particular, the clocks on system elements will gradually drift apart, and thus a difference of a few seconds between events recorded by different machines cannot be considered significant. However, the accuracy will be ample for events measured in human terms.

Note also that this facility (or one based on it) would make a reasonably reliable unique name generator for whatever purpose one is required. In particular, the probability is low that a name (time-value) will be generated on a particular system element more than once at intervals greater than the resolution of the clock. Similarly, the probability is high that a malfunction which might lead to duplicate names being generated will be detected rapidly because of the high user visibility of the time.

Data Formats

We propose that the time be represented in forms similar to those used by the existing Alto Mesa timing facilities (see the module **TimeDefs.Mesa**). In particular, the *packed* format of the time will be a 32-bit cardinal recording the number of seconds since an arbitrary base time. Note that this counter overflows in approximately one century, so that January 1, 1901, is *not* a suitable base time for our product. The *unpacked* format is a record containing cardinals representing the year, month, day, hour, minute, second, and possibly the day of the week and "time zone." Because watch chips typically return the time in this format, Pilot will probably do so also (unless we get a very strange one!).

Pilot Functions

Pilot will provide just two operations for time of day:

GetTimeAndDate:- Returns Greenwich Mean Time in the Unpacked format.

AdjustTimeAndDate:- Accepts an integer argument indicating the error (in seconds) of the watch chip from the "correct" Greenwich Mean Time.

The operation **GetTimeAndDate** will be a public procedure available to all Pilot clients. The operation **AdjustTimeAndDate** will be provided only to *bona fide* timer maintenance software and will not be available to other clients. This minimizes the need of Pilot clients to monkey with the time and hence the chance to get it wrong.

Common Software Functions

Common Software should provide at least the following facilities to support the time of day clock:

UnpackDateAndTime:- accepts a value of the time in packed format and returns it in unpacked format.

PackDateAndTime:- the inverse of **UnpackDateAndTime**.

GetLocalTime:- accepts a value in Greenwich Mean Time (packed format) and returns the corresponding value in local time, adjusted for Daylight Savings Time, etc. (should it also take a parameter stating the "time zone" or should this be derived from global data?)

GetGMT:- inverse of **GetLocalTime** (is this really necessary?).

AppendDateAndTime:- accepts two arguments, a string and an unpacked value of time, and appends a printable representation of the latter to the former; this might be subdivided into operations which separately append the date, time, day of the week, and local "time zone" (such as EDT, PST, etc.).

WaitUntilTime:- accepts an argument representing the Greenwich Mean Time at which operation returns; i.e., the calling process is put to sleep until then.

Daylight Savings Time adjustments should be made automatically by Common Software. I.e., the user should not have to perform a separate operation to set the clock ahead or back. Furthermore, the adjustment should always be relative to the actual time being converted.

E.g., a GMT time stamp stored with a file of, say, 1200, July 1, 1977, would be converted as 5:00 AM Pacific Daylight Time, independent of whether the conversion takes place in the summer or the winter.

Installation Parameters and Multi-national Considerations

Because the processor maintains only Greenwich Mean Time, the Common Software facilities must include an installation parameter to account for the local time and time zone. These facilities must also reflect local and national conventions in presenting the time, changing to Daylight Saving Time, etc. Thus, the following are the installation parameters which must be settable by the Xerox installation representative (or possibly the customer himself).

The **AppendDateAndTime** routine:- This must convert the date and time to a string in the language and format of the customer; i.e., day first or month first, etc.

The local time zone.

A Daylight Saving Time adjustment routine, based on either an algorithm or a table.

The Base Time for packed format.

Setting the Time

Clearly, if the hardware time of day clock is going to drift, then the system must provide some way of correcting it. Furthermore, the customer may also need to change the relationship between his local time and Greenwich Mean Time (due to error, relocation, change in law or custom, etc.). I suggest that these two problems are separate and should be treated separately. In particular, the problem of the relation between local time and GMT is treated the the previous sections.

This leaves the question of adjusting the hardware clock to compensate for drift, battery failure, or malfunction. I propose that any facility along these lines be implemented as a separate application program which is called only as circumstances require. There are a number of different methods which might be used, depending upon the system configuration and the options the customer is willing to buy. These include:

For stand-alone system elements, a program which asks the user to indicate how far forward or back to adjust the clock.

For system elements on a Xerox Wire, a program which polls other system elements and compares its clock with theirs. It sets its clock to the median (*not* the mean) of the others if it is wildly different from them.

For Xerox Wires or other OIS networks which have one or more server machines with clocks officially designated as "accurate," a program which checks the local clock against the official clock server.

For those who need to know the time very accurately, a hardware device which derives the national standard time from an external source.

For those who need to know it fairly accurately but only occasionally, a dial-up time server maintained by Xerox, the phone company, or the government, and which quotes the time in a standard digital format.

It is an option depending upon the customer's requirements and the system and network configuration whether the time correcting program is run manually on demand or automatically at preset times. It is an independent option whether or not it interacts with the human user and asks for confirmation.

Distribution:
Lynch
Shultz
Liddle
Metcalfe
Stottlemyer
Wick
Irby
Kimball
Johnsson
Redell
McJones
Gifford
Horsley
Bishop
Rosen
Thacker
White
Dalal
Murray
Schwartz
Jarvis
Clark
Reber