Inter-Office Memorandum

To Bill Lynch

Date October 14, 1977

From H. C. Lauer

Location Palo Alto

Subject Miscellaneous Comments on Pilot File System

Organization SDD/SD

XEROX

Filed on: <Lauer>files.memo, .ears

The following are a set of miscellaneous thoughts and recommendations on the Pilot File System. They range from minor design and implementation questions to non-trivial strategy and planning issues.

1. Maximum File Sizes

Butler has suggested that the maximum size of a file in Pilot be limited to 2^{25} bytes or 2^{16} pages. The principal advantage of this is that it permits file page numbers to be represented by a single word. Since these will be liberally stored throughout most kinds of files, their size has a significant impact on the formats of those files. It will also be simpler to manage them in application software. The resulting limit imposed upon file sizes would thus be 32 Megabytes, large enough for almost all requirements. Those applications which would not be satisfied by this limit would probably be better designed to use multiple files anyway.

Recommendation: Implement this lower limit.

2. Extended Attributes

Butler also recommended that we reconsider the representation of the extended attributes of a file. In particular, identifying the attributes by CARDINALS and representing them by unspecified words is not very satisfactory or typesafe. We ought to consider instead some means of registering attributes with Pilot and giving them more structure. The kind of information which we ought to keep in the file attributes rather than in the file itself is that information which has a broad application across Pilot rather than being relevant to just one subsystem.

In particular, one attribute which ought to be included from the very beginning is an "object type" which indicates what kind of file it is and hence what systems (and versions of systems) are required to interpret it.

Recommendation: Redesign the attributes and invent a registration mechanism for attributes and object types, preferably prior to release of Pilot 2.0.

3. Redundancy and Scavenging

Peter Bishop recommends a scheme of recording in the header of each file page certain information describing that page. E.g., this would certainly include the file identification and the page number within that file. It might also include other

information as appropriate. This would be useful in scavenging the volume and in other forms of recovery from errors, crashes, and problems.

Recommendations:

a. Design this header block as part of the file system.

b. Specify an interface (probably private) for getting access to it.

4. Volume File Capabilities

Volume File Capabilities seem to me to serve no useful purpose. The control, management, and distribution of them is likely to cost more--both in implementation and at run-time--than any benefits which might accrue. In particular, since protection is not one of the features of Pilot, the 'protection' afforded by capabilites to identify volumes is more apparent than real. A volume is really too coarse a grain to apply prohibitions against creating or deleting files. Those limited cases in which the file space must be managed (e.g., in a central file server) can be handled by client software built for the purpose. In any case, deleting files can be controlled on a file-by-file basis with the *delete* permission in the file capability.

Recommendation: Delete Volume Capabilities and return to the simple Volume IDs and we had previously.

5. Swapping over the Xerox Wire

Swapping over the Xerox Wire is a bold idea and an important experiment.

The *pay-off* of this experiment is, in my opinion, enormous. It will give us tremendous flexibility in configuring Office Information Systems at lower cost, in developing specialized products for specialized markets, and in enabling the design of future low-cost and stripped down models of our equipment. I see it as a major market advantage beyond the scope of anything which can be quantified.

The *risks* are equally great and must be managed carefully. Since nothing like this has been tried before, we are taking a chance in doing it at product development time rather than in the research lab. Whatever we do is likely to need a lot of engineering and reengineering before it can be released to the public. Furthermore, it could fail altogether, either in capability, performance, or system structure.

To minimize these risks, I strongly support the position that:-

a. *We restrict X-Wire swapping to within one campus.* The delays likely to ensue as a result of swapping to some distant host over arbitrary gateways and communication links will throw unpredicatable anomolies in the performance of Pilot and its clients. These can hurt us very badly even if the code works correctly. We are not in a position to experiment with all of the permutations and combinations likely to occur in the field with real customers, and we should therefore restrict our attention to situations we can control.

b. *We swap only to designated, dedicated server system elements.* The role of a swapping host could very well be a full-time job for a D0, depending upon its environment. We should only add other functions if there is spare capacity available; in particular, we should not regard the swapping function as something which can be added at random to other system elements with other responsibilities. Most important, we should not plan at all to swap between arbitrary system elements because the problems of managing both the load and the availability will overwhelm us.

In any case, getting a viable Xerox-Wire swapping mechanism is likely to take a lot more time than we can reasonably predict now.

Recommendation: We should include in our work plan experiments to try swapping over the Ethernet from an Alto Pilot environment at the earliest possible time.

6. Pilot and Clearinghouses

The above assumptions about accessing remote files will help us straighten out the relationship between Pilot and a clearinghouse for locating files. Given a capability for a file, Pilot will follow some variation of the following scenario to access that file:-

a. It will interrogate the volume file maps of any volumes which are attached to that system element; if the file is found, it will be accessed there.

b. It will interrogate its designated swapping server(s) using a part of a *swapping protocol;* if a server has the file available, Pilot will access it via the swapping protocol from there.

c. Otherwise, Pilot will return a *non-resumable* signal to the client that the file cannot be found.

Note that there is no future in the client or its clearinghouse trying to resume such a signal because Pilot would not be able to access the file anyway (since it is not on a swapping server). Instead, the client would have to invoke FTP or something else to get a copy made where is can be accessed and then try with a new capability. Thus the notion of a clearinghouse is strictly a client software notion (along with such things as directories) and there need be no special interface to Pilot to supply information about the location of files.

Of course, once a file is located, the relevant information would be cached by Pilot (*and by the swapping server*) for rapid access in the future.

Recommendation: Delete all reference to clearinghouses from the Pilot Functional Specifications.

7. Problems to be Resolved about Universal File IDs

There are still a number of issues to be resolved about the scheme for have universal file identifiers as means for identifying files. Among this issues are:-

Size: I do not agree with Peter Bishop's assessment that 64 bits is enough. In particular, he assumes a 20-bit processor serial number. This does not permit any scope for structure in serial numbers, as there undoubtedly will need to be before we are done.

Generation: There are currently no plans to include a non-volatile memory in the D0, and I understand that it is too late to incorporate one for the first release. Since all of the UID schemes described to date require such a memory, some interim measure must be taken.

Maintenance: All of the schemes have the possibility of overflowing the UID space in some plausible cases, requiring a maintenance operation on the part of Xerox. I am strongly against such backup maintenance plans on the grounds of cost and clumsiness: On the one hand, we will have to build into our maintenance and training programs the

knowledge and apparatus for reassigning some identifier. One the other hand, the need to do this will occur sufficiently infrequently that the skills necessary in the maintenance organization to recognize and service this situation will be forgotten. Both of these will force on Xerox (in particular, upon Xerox branch offices) random costs which are disproportionate to the return.

Binding: Given that Pilot and its clients refer to files by these Universal File IDs, we must consider how they discover the correct capabilities, particularly for immutable files which will be distributed and redistributed frequently. It seems to me that some combination fo the following two possibilities will be used:

a. A program wishing to access a file dynamically interrogates a directory which maps some symbolic identification of that file into a capability. In this case, the fact that the file ID is unique over all OIS systems is irrelevant (i.e., the directory could just as well have pointed to the location of the file directly), and we have imposed an extra level of map upon the client.

b. A program wishing to access files remembers internally (over a long period of time in some file somewhere) the correct capabilities. In this case, we have two binding problems which must both be done in the field. First, when a file is replaced with a new version, all systems which refer to the old version must be located, and for each it must be determined whether or not to replace its old capability with the new one. Second, the new file must be given capabilities for the appropriate verions of the files it refers to. This binding problem is a generalization of that of initializing Bravo. However, the process will be highly visible to the customer in Peoria, and both costs and Xerox's image could get out of control very quickly.

There has, as far as I can determine, been no coherent strategy for undertaking this binding.

Recommendation: Think very hard about these problems before we fall into a pit.

c: Liddle
Shultz
Redell
McJones
Horsley
Gifford
Bishop
Moore